Jan Camenisch
Ronald Leenes
Dieter Sommer (Eds.)

Digital Privacy

PRIME – Privacy and Identity Management for Europe

Springer

Lecture Notes in Computer Science 6545

Jan Camenisch   Ronald Leenes   Dieter Sommer
(Eds.)

Digital Privacy

PRIME – Privacy and Identity Management
for Europe

 Springer

Volume Editors

Jan Camenisch
Dieter Sommer
IBM Research
Säumerstr. 4, 8803 Rüschlikon, Switzerland
E-mail: {jca, dso}@zurich.ibm.com

Ronald Leenes
Universiteit van Tilburg
TILT - Centrum voor Recht, Technologie en Samenleving
Postbus 90153, 5000 LE Tilburg, The Netherlands
E-mail: r.e.leenes@uvt.nl

Foreword

During the past decade the digital society has firmly established itself in Europe and in many other parts of the world. Information and communication technology has emerged as the mission-critical backbone of modern economic and social life in the early 21st century. The contours of the dawning digital age have taken shape, and they include massive amounts of data, personal and non-personal, being continually generated and ceaselessly processed, exchanged, recombined, and often stored for indefinite periods of time. In this context, the creation of an electronic identity management infrastructure that puts the management of digital identity data under the users' control has manifested itself as one of the central challenges for life in the digital age.

The Internet transcends geographic and jurisdictional borders; hence there is a strong need for a global approach to trust in the digital society. Citizens look for value in the activities they do on the Internet. They want to be able to trust the technology and services provided and the actors behind. To forge a just and dignified digital future, and considering that trust is subjective and depends on context and culture, we need digital means, tools and instruments to allow us to sense the variables and reach a conclusion on the trustworthiness of services and third parties prior to engaging in interactions. To contribute to meeting these demands of the new age, trust, and with it identity management and privacy protection, are prominent elements of the ICT Research and Development Framework Programme of the European Union.

A flourishing digital society expects diversity, usability and openness, interoperability and competition as key drivers for trust and security. Diversity reduces the risks coming with dependence on one type of technology, and open standards and interoperability are key to competition, to empowering users to choose among a variety of products and services, and to the creation of business opportunities for small, medium and large companies alike.

PRIME – the 2008 recipient of the internationally renowned IAPP award for the best contribution to innovative privacy technology – has put Europe on the global map as a place for high-quality research on privacy. It has

effectively contributed to the preparation of Europe for a new digital age, for a digital life that preserves the shared European values of democracy, freedom and civil liberties. We would like to thank the PRIME project and all its partners for this opportunity to draw attention to the European Commission's efforts in this domain, and for putting Europe on the map as a global thought leader in privacy protective digital identity management.

December 2010   Jacques Bus (Head of Unit)
                                                  Dirk van Rooy (Head of Sector)
                                               DG Information Society and Media
                                                         European Commission

Preface

Information technologies are becoming pervasive and powerful to the point that the privacy of citizens is now at risk. Indeed, more and more of our daily transactions are conducted electronically and require us to transmit personal information. Examples include using an electronic identity card to prove one's age in a bar, buying digital content on the Internet, checking our healthcare records on-line, or planning our next vacation. In this new information society, individuals need to be able to keep their autonomy and to retain control over their personal information, irrespective of their activities. The widening gap between this vision and current practices on electronic information networks undermines individuals' trust and threatens critical domains like mobility, healthcare, and the exercise of democracy.

Why Privacy and Identity Management

Closing this gap requires an identity management system that puts the users in control of their data and allows them to protect their privacy in electronic transactions. Indeed, we all manage our personal information (and thereby our identities) in our daily lives. However, the way we have learnt to do so for our non-electronic lives works poorly in the electronic society now taking shape for a number of reasons. First, we are often not aware what data about ourselves we are revealing in a transaction or we might even not be aware of the fact that we are revealing data to start with (e.g., making a call with a mobile phone reveals all kinds of (unexpected) data to unexpected parties). Second, the sheer complexity of the applications and their building blocks makes it almost impossible to understand where our data flows. Third, even if we were capable and willing to manage our electronic personal data and identities and protect our privacy, we would usually not be able to do so because the applications don't allow us to do so due to the way they are built. A well-known example is that users were asked for their social security number just so that the application could use it as a unique identifier.

PRIME's Solution

We, the PRIME partners, noted that the state of the art in privacy-enhancing mechanisms provides the technical means to build such a privacy-enhancing user-centric identity management system that would empower the users to manage their identity and protect their privacy. Thus the PRIME consortium was formed to prove this and to raise the awareness for privacy issues and their solutions. We chose an integrated approach to the legal, social, economic, and technical areas of concern to research, develop, and evaluate solutions to privacy-enhancing user-centric identity management. During the course of the project, we have developed a framework that integrates all technical and non-technical aspects of privacy-enhancing identity management and shows how privacy-enhancing technologies can be employed to realize privacy-enhancing user-centric identity management. We have elicited detailed requirements from legal, social, economic, and applicational points of view and have shown how they can be addressed, i.e., how to enable the users to effectively control their private sphere. That is, we have put forth an architecture that orchestrates the different privacy-enhancing technologies, including the human-computer interface. Based on this architecture, we have built several prototypes that exemplify privacy-enhancing user-centric identity management for a few selected application domains. We have validated our results by conducting experiments with end-users in these application areas. Moreover, we have considerably advanced the state of the art to address foundational technology, through research on human-computer interface, ontologies, authorization and cryptology, anonymous communications, and privacy-enhancing identity management systems architecture and assurance methods.

This Book

This book reports on the findings of the PRIME project. It is partitioned into five parts. The first part is a summary. It explains the privacy issues based on the example of Alice who goes shopping in the Internet. It then explains how PRIME resolves these issues. The second part of the book provides the legal, social, and economic landscape of privacy-enhancing identity management and derives requirements for privacy-enhancing user-centric identity management. The third part explains how privacy-enhancing user-centric identity management can be realized. It first describes the PRIME architecture which brings together the different privacy-enhancing mechanisms. These mechanisms are then subsequently explained. That is, we not only describe the results that the project has obtained based on these mechanisms, but also give a comprehensive overview of these technologies. The fourth part reports on how the PRIME architecture can be applied to applications. It describes the application prototypes that we have implemented and the lessons

we have learnt. This part further summarized the requirements on privacy-enhancing user-centric identity management in general and how they can be addressed. The fifth part features the conclusions we have drawn, provides an outlook to the future of trust, privacy and identity management, highlights open problems, and describes PRIME's follow-on projects.

More details for the results covered in this book as well as additional materials are available on the PRIME website www.prime-project.eu. The next page contains an overview of these materials.

July 2008
Jan Camenisch
Ronald Leenes
Dieter Sommer

Available PRIME Materials

The following materials are available on www.prime-project.eu:

Introductory Documents:

- Press releases, leaflets, and slide presentations outline the project objectives, approach, and expected results;
- The PRIME White Paper introduces privacy-enhancing identity management issues and PRIME's vision, solutions, and strategies;
- A number of tutorials introduce major concepts of privacy-enhancing identity management for use by the software development community and the general public.

PRIME Technical Materials:

- PRIME Framework reviews privacy-enhancing identity management issues; PRIME legal, social, and economic requirements; PRIME concepts and models; and PRIME architecture outlines.
- PRIME Requirements analyze in depth the legal, social, economic, and application requirements. They comprise generic requirements, as well as specific, scenario-based requirements of selected application areas including eLearning, location-based services, and airport security controls.
- PRIME Architecture describes in depth the organization and orchestration of the different privacy-enhancing technologies in a coherent PRIME system.
- Annual Research Reports review the research results gained in PRIME in each of the four years, and the research agenda for the subsequent years.
- HCI Guidance provides a comprehensive analysis of the Human-Computer Interface requirements and solutions for privacy-enhancing identity management.
- Assurance Methods survey the existing assurance methods that are relevant to privacy-enhancing identity management.
- Evaluation of Prototypes assesses a series of early PRIME technology prototypes from legal, social, and economic standpoints.
- More than 200 scientific publications address results produced in all PRIME-related fields within the scope of the project. The abstracts of those papers and links to them are listed in the four Annual Research Reports which are available from PRIME's website www.prime-project.eu.

Project Partners

The PRIME Project was undertaken by the following 22 partners.

Compagnie IBM France	France
International Business Machines of Belgium	Belgium
IBM Research GmbH	Switzerland
Unabhängiges Landeszentrum für Datenschutz	Germany
Technische Universität Dresden	Germany
Deutsche Lufthansa AG	Germany
Katholieke Universiteit Leuven	Belgium
T-Mobile International AG	Germany
Hewlett-Packard Ltd.	UK
Karlstads Universitet	Sweden
Università degli Studi di Milano	Italy
Joint Research Centre	Italy
Centre National de la Recherche Scientifique	France
Johann Wolfgang Goethe Universität Frankfurt	Germany
Chaum LLC	USA
Rheinisch-Westfälische Technische Hochschule Aachen	Germany
Institut EURECOM	France
Erasmus Universiteit Rotterdam	The Netherlands
JaTeK GmbH	Germany
Universiteit van Tilburg	The Netherlands
Fondazione Centro San Raffaele del Monte Tabor	Italy
Swisscom AG	Switzerland

Of these partners, JaTeK left the consortium shortly after the project had started as the company dissolved, IBM France was replaced by IBM Belgium because of the change of the project coordinator, and Institut EURECOM left in the third project year because of key persons leaving the institution.

Contributors to PRIME

The following people contributed to the PRIME Project.

Gérard Lacoste from IBM France; Eric Goderniaux from IBM Belgium; Endre Bangerter, Anthony Bussani, Jan Camenisch, Günter Karjoth, Susan Hohenberger, Abhi Shelat, Dieter Sommer, Michael Waidner, and Roger Zimmermann from IBM Research; Marit Hansen, Marita Häuser, Henry Krasemann, Christian Krause, Jan Möller, Antje Nageler, Maren Raguse, Martin Rost, Jan Schallaböck, and Harald Zwingelberg from Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein; René Balzer, Mike Bergmann, Stefan Berthold, Katrin Borcea-Pfitzmann, Rainer Böhme, Alexander Böttcher, Sebastian Clauß, Hilko Donker, Elke Franz, Andreas Juschka, Benjamin Kellermann, Thomas Kriegelstein, Katja Liesebach, Andreas Pfitzmann, Stefanie Pötzsch, Immanuel Scholz, Stefan Schiffner, Anne-Katrin

Acknowledgements

Contents

Part III: What Technology Can Do for Privacy and How

**19 Reputation Management as an Extension of Future
Identity Management** 557

Sandra Steinbrecher, Franziska Pingel, and Andreas Juschka

20 Human-Computer Interaction 569

*Simone Fischer-Hübner, John Sören Pettersson, Mike Bergmann,
Marit Hansen, Siani Pearson, and Marco Casassa Mont*

Part IV: PRIME Applied

Part I

Privacy and Identity Management

1

An Introduction to Privacy-Enhancing Identity Management

Jan Camenisch[1], Ronald Leenes[2], Marit Hansen[3], and Jan Schallaböck[3]

[1] IBM Research – Zurich
[2] Tilburg University
[3] Unabhängiges Landeszentrum für Datenschutz

The PRIME project demonstrates the viability of privacy-enhancing identity management. By this we mean identity management solutions that manage the individual's identity online and that also empower the individual to actively protect their own privacy.

The guiding principle in the PRIME project is to put individuals in control of their personal data. The notion of user control has been adopted in many recent user-centric identity management initiatives.

However, most of these initiatives only takes the first steps on the way to a new generation of identity management systems. They do not provide adequate safeguards for personal data and are limited in giving individuals control over their personal data. Effective management of information privacy requires new tools starting with the minimisation of personal data disclosure. Furthermore, users can be empowered with tools that allow them to negotiate privacy policies with service providers. This would require systems that enforce agreed policies by technical means, and keep track of data collection and usage. In addition to user side applications, service providers will be required to put adequate protection mechanisms in place and align business processes to take advantage of these mechanisms.

1.1   Motivation

The internet, by design, lacks unified provisions for identifying who communicates with whom; it lacks a well-designed identity infrastructure.[1] Instead, technology designers, enterprises, governments and individuals have over time developed a bricolage of isolated, incompatible, partial solutions to meet their needs in communications and transactions. The overall result of these unguided developments is that enterprises and governments cannot easily identify their communication partners at the individual level. Given the lack of a proper identity infrastructure, individuals often have to disclose more personal data than strictly required. In addition to name and address contact details such as multiple phone numbers (home, work, mobile) and e-mail addresses are requested. The amount and nature of the data disclosed exceeds that usually required of real world transactions, which can often be conducted anonymously – in many cases the service could be provided without any personal data at all. Over the long run, the inadequacy of the identity infrastructure, that takes the above into account, affects individuals' privacy. The availability of abundant personal data to enterprises and governments has a profound impact on the individual's right to be let alone as well as on society at large. The online world is a complex new environment. Social structures online have to be established within a short time - very much unlike their real world counterparts. At first glance those procedures based on personal contact or paper are transformed into digital procedures for use online. But below the surface, more fundamental differences between the offline and the online world exist, such as the relative permanence of memories and the ease with which experiences can be shared between many of actors across time and space barriers.

We are beginning to understand that these differences are both qualitative (e.g., automated decision making) and quantitative (e.g., more data collected and stored for a longer period) in nature. The speed of developments and potential irreversibility of their effects requires urgent attention on issues such as identity, trust, security, and privacy.

The – sometimes conflicting – interests and issues that have to be reconciled are increasingly well understood. For example for such a conflict is an interest in identifying trading parties on one hand and providing anonymity on the other. The convenience of 'portable' online identities is another example; users do not want to fill in similar forms for each service, yet there is the risk of disclosing more than is required. National security interests – sometimes positioned as overriding civil liberties in public debates – increases the need for proper data protection. And finally, while customer data is an

---

[1] The Internet has an identity infrastructure often identifying only the endpoint of a communication: IP addresses. These are often unreliable to identify users.

important business asset, they can become a business liability in complying with data protection legislation.[2]

Online *identity management* is in need of reconsideration. The patchwork approach to online identity needs to make way for a more elaborate design that takes into account the various stakes and issues. Indeed the identity management landscape appears to be changing. *Enterprise identity management* is slowly making way for *user-centric identity management*. Various initiatives, such as the Liberty Alliance project and WS-Federation, aim to pave the way for identity management that 'involves the users in the management of their personal information and how that information is used, rather than to presume that an enterprise or commercial entity holds all the data' [LAP06]. Establishing authenticated individual identities within and across organisational boundaries are the primary business drivers behind these initiatives. Their successful adoption depends on improved privacy protection. User control and other elements of privacy protection also gain attention in a broader sense. The '7 laws of identity' [Cam05] initiated by Microsoft's Kim Cameron clearly attracted attention in the identity community.

What these developments show is that industry is adopting the notion of user control over personal data. But so far the interests of the service providers are better served than those of the individuals. In the wake of what is coined Web 2.0, where consumers merge into prosumers, services replace applications, data increasingly drives economic activity, and where generally the landscape becomes more dynamic, this will not do. Individuals will feel a stronger desire for privacy and control over what's known about them. They also require more security, which demands stronger and better authentication and identification, which in turn requires even better privacy protection.

The PRIME project aims to show that seemingly disparate notions such as anonymity and accountability, security and privacy, and informational self determination and enterprise needs can be reconciled. PRIME intends to set the boundaries for the emerging identity management infrastructure with a clear balance of the interests of users, enterprises and society.

The PRIME project takes the perspective of the individual and places the individual at the core of user-centric privacy-enhancing identity management. This leads to a different, but not incompatible, set of requirements. The requirements elicited in this document have their roots in the OECD Privacy Guidelines [Org80], the Council of Europe's Convection No.108 [Cou81], the Fair Information Practices (for instance embedded in the CSA privacy code [Ass]), the EU Data Protection Directive (Directive 95/46 EC) and recent discussions on user-centric identity management. Many of these requirements are discussed in Kim Cameron's '7 Laws of Identity' [Cam05] and the

---

[2] This is particularly so in the 44 states (as of July 2008) in the US that have enacted Security Breach Notification Laws. These laws require companies to report security and privacy breaches which could subsequently result in liability cases and damage of reputation.

Ontario Information and Privacy Officer's white paper on identity management [Cav06].

1.2   A Scenario – Alice Goes Shopping

The requirements elicited in the following pages may appear to be ambitious, but software prototypes that demonstrates these features have been developed and evaluated within the PRIME project. Before looking at them in more detail and describing the PRIME approach to address them, we will first take a walkthrough current practice and the problems it entails in a typical online shopping scenario. The purpose here is to showcase the software architecture required for enabling privacy-enhancing identity management to those organizations interested in deploying these features. Figure 1.1 illustrates the exchange of personal data in a typical online shopping scenario today. Alice asks her sister Alicia, whom she dearly trusts, for advice on white wine. On the basis of her sister's recommendation, she orders a box of bottles of Chardonnay at CyberWinery. To this end, Alice has to provide personal data (name, delivery address, and possibly payment data, such as her credit card data). If this is her first and only order, the winery will store only some of these data in their records. But more likely, it will ask Alice to register, arguing that this will make it easier for her to make additional purchases. If she does, Alice will have an account at the winery which not only contains her name and address, but also her purchase history, personal preferences, and likely also her credit card data.

Suppose the winery has outsourced warehousing and delivery to LogisticsProvider, a major logistics company. LogisticsProvider needs to have some of Alice's personal data – name and shipping address – to deliver her order. CyberWinery checks Alice's credit card details at CreditProcessor for credit authorisation. If the order is accepted, CreditProcessor also takes care of processing the payment. Again, Alice's personal data are exchanged between two businesses. CreditProcessor will store transaction details in their records for business and accountancy purposes.

Suppose Alice also takes up Alicia's recommendation to purchase the Ultimate Wine Guide at CyberBooks, *the* online bookstore for Gourmet books. She again has to register before being able to order, and she basically has to provide the same information she provided to CyberWinery. Consequently, the CyberWinery scenario unfolds again, most likely involving CreditProcessor, and possibly even involving LogisticsProvider as well.

The scenario sketched encompasses many exchanges involving personal data between user and service provider and between service provider and their associates. Many of these data are stored in multiple databases. Some providers can make interesting inferences on the basis of the data they have. CreditProcessor, for instance, knows where Alice does her shopping and the amount she spends, whereas LogisticsProvider even knows what she buys and
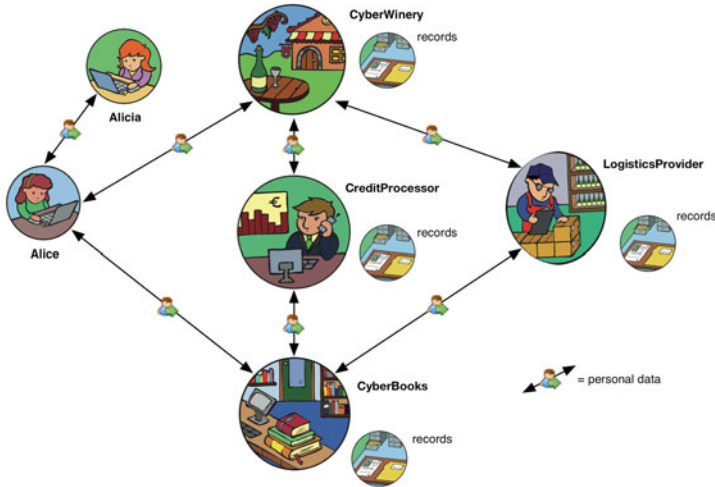
Fig. 1.1 User data exchange

where. CyberBooks has a much dimmer picture of Alice's shopping habits, they only get to see what they contribute to Alice's collection of cook books.

Overall, this scenario illustrates a number of issues from a user's perspective, especially if she wants to minimise the risk that her data may be abused, for instance for identity fraud, or for profiling and *social sorting.*

Many of these problems can be addressed by means of novel identity management systems. In this paper we discuss various problems and describe the way the PRIME project aims to resolve them by offering a privacy-enhancing identity infrastructure. We will use Alice's online shopping scenario to unravel the problems and formulate a list of requirements on our way.

1.3 PRIME Enabled Shopping

The aim of the PRIME project is to provide privacy-enhancing identity management tools for individuals. PRIME empowers the user by offering them more extensive *(user) control* over their personal data. The *PRIME toolbox* offers support for creating, using and keeping track of multiple digital identities and the (certified) attributes associated with them. It allows (certified) attributes to be transferred between entities, such as user and service provider, or between service providers. It also extends the user's control over attributes disclosed to remote entities. The PRIME vision is based on the principle of data minimisation, i.e., disclosing and processing personal data only to the extent necessary. To limit the transfer of personal data for authentication purposes, claims and credentials are used to establish trustworthy relationships.

Where necessary, for instance for certifying certain user attributes, PRIME makes use of privacy-enhancing public key infrastructures and trusted third parties. The integrity of claims in PRIME enabled communication is guaranteed by cryptographic techniques. Each party in the interaction makes use of *PRIME Middleware.* The individual users additionally use the *PRIME Console* to manage their personal data. User applications (such as web browsers) may delegate identity management tasks to the PRIME Console and PRIME Middleware. The trustworthiness of the PRIME components should be maximised by technical means (e.g., cryptographic techniques) and non technical means (e.g., certification and assurance). We will now explore Alice's ventures in the online wine business by going through six phases to illustrate online transactions from before entering the internet to becoming a frequent shopper and beyond.
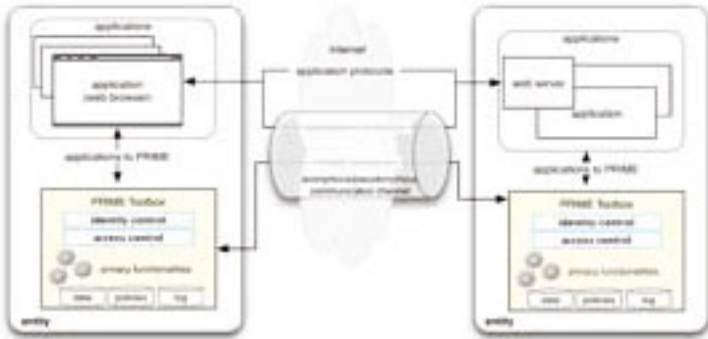
Fig. 1.2 PRIME's high level architecture

1.3.1   Phase 1: Buyer Beware

Transactions require a certain level of mutual trust between transaction partners. Each party has to be confident that the other will perform their contractual obligations, will not abuse one's vulnerabilities, and that there are options for redress in case of breaches. In the offline world this confidence stems from factors such as the respect commanded by the brick and mortar that houses (commercial and governmental) institutions and from honourable social institutions such as the legal system that acts as a safety net in case of conflicts.

In the online world tangible signs of trustworthiness are absent to a large extent and therefore we have to rely on other signals for trustworthiness. It is relatively easy to create websites that resemble genuine ones. This method is therefore frequently employed by criminals for all kinds of fraud (including ID

theft and phishing attacks). Although people may believe a certain website to be genuine, a reliable level of trustworthiness cannot be established. Assurance that a service provider is genuine and complies to regulations and policies can be provided by third parties (trust assurance), however, some users have difficulties in understanding the scope and value of these trust marks.

In addition, the communication channel needs to be trustworthy because communication can be intercepted, manipulated and suppressed. User provided data, such as credit card data can, when in the hands of the wrong people, have serious implications for the user. Integrity and confidentiality of both communication and data are therefore important requirements for online interactions. The users should substantially be able to trust the entire chain of entities involved in providing a service to be secure against intruders, eavesdroppers, etc - or better: not even need to rely on trust but stay in control. This calls for technical measures, such as, encrypted and properly authenticated communication.

Additionally non-technical measures can increase user confidence that their interactions are safe. Transparency, i.e., providing clearly understandable information to the user on the data processing, is an example. Online processes – like shopping or simply gathering information – are currently rarely transparent and many users do not feel comfortable because of the technology involved in the interaction. Prospective customers often even have to guess or do not understand what the shopping process will look like when engaging in it.[3] Improving the transparency of the processes and making clear why personal data are being collected and what happens with the data makes users feel more comfortable in online interactions and helps to build their trust.

Based on her sister's recommendation, Alice decides that it may be worthwhile looking for wine at the CyberWinery. The store implements a number of measures that reassure Alice of its trustworthiness. CyberWinery's home page shows a trust mark she is familiar with and that she considers trustworthy. The shop also turns out to be PRIME enabled, which means that she knows how the communication will work because it is well documented and she has experience with it. For the shop having a PRIME enabled customer means that it is able to check the validity of certain credentials provided by this customer by means of trusted third parties (see Phase 3).

1.3.2   Phase 2: Pre-sales — Starting from Maximum Privacy

Alice's online interest in white wine does not appear to be particularly private or sensitive, when compared with her visits to health insurance websites or medical websites that might reveal information she wants to keep private. However, incorrect and damaging inferences may be drawn from Alice's wine interest when disclosed at the wrong place at the wrong time. Her search for

---

[3] Despite legal requirements (e.g., the e-Commerce Directive 2000/31/EC), many online shops still do not offer clear documentation of the shopping procedures.

wine during working hours may reflect a drinking habit, even though it just happens to be that she is organising a cocktail party for a colleague. Alice is sensitive about disclosing data that may lead to the wrong conclusions, about leaving online trails about her online transactions and she may even be worried about identity fraud due to recent newspaper reports. She guards her private sphere and wants to remain as unobserved as possible. She adheres to the *data minimisation* principle and starts her online journeys from maximum privacy, choosing to disclose more personal details with her consent and according to the her own preferences.

Alice studies the company's general privacy policy. The shop has implemented the Article 29 Working Party's recommendation of layered policies [Art04]. The shop's home page shows the simple and short outline of the privacy policy and offers a click through to more detailed explanations of the company's policies. The privacy policy states CyberWinery's intentions regarding the protection of personal data. It assures the user that the data obtained by the store during browsing, purchasing and later on for delivery (see Phase 3) will be handled as stated in the policy and will only be made available on a need to know basis. Alice is assured for now that the shop meets some basic requirement (see *justifiable parties*). The policy also states that the shop will allow her to opt out of their direct marketing programme at all times if she cares to join it. It also explains that her IP address is only recorded for statistical purposes, but not for profiling her behaviour.

IP addresses warrant caution because they are in many cases identifying data, albeit not very reliably.[4] They are like breadcrumbs left behind as a trail of the user facilitating linking her behaviour from one site to another. Due to the inadequacy of IP addresses as identifying data, they are sources of false conclusions about internet users. The principle of data minimisation can be applied to IP addresses as well. The shop should refrain from storing them unless there are legitimate reasons to store them. Alice can use an anonymising service, such as TOR or AN.ON, to hide her IP address from the webshop. This would reduce her concerns about leaving IP breadcrumbs. The PRIME Middleware provides interfaces to such anonymising services which makes it easier for the user to use these services.

1.3.3   Phase 3: Ordering — Informed Consent and Purpose Limitation

Autonomy as a central concept implies that individuals should make their own choices and only be bound to contracts they knowingly and voluntarily enter into.[5] As there usually is an asymmetry in both power and information

---

[4] IP addresses may be shared by multiple users, e.g., multiple PCs behind a firewall, cybercafes, dynamic IP addresses distributed by ISPs.

[5] Of course individuals also have legal obligations vested by the State they may not subscribe to voluntarily or enthusiastically, but even here they can voice their choices in elections.

to the detriment of the individual, it is reasonable to protect the individual in their relation with enterprises and governments. To this purpose regulation obliges service providers to state who they are and what their terms and conditions are, and what the effects of contracts they enter into are. This allows individuals to make informed choices and also provides them with information they need if they seek redress in case of contractual breaches, problems, and so forth.[6] The information requirements also apply to the collection and use of personal data because this affects the individual's privacy.

When Alice decides to purchase a box of white wines she must disclose some personal data in order to complete the purchase order. To determine which data are reasonable to disclose, she has to dig deeper in the shop's general privacy policy requiring serious effort. She has to consider the information the shop is obliged to provide about the purpose of data collection, the duration the data are kept, etc. On the basis of this information, she may decide that, in her opinion, certain data is excessive and she may decide to proceed, not to proceed, or provide false data. Assessing privacy policies is not easy in current environments. Many general privacy policies state the website's policy in lengthy difficult language that appears to show that the website really has considered all the intricacies of online transactions rather than providing the customer with relevant information. They are generally not written with the average user in mind. Although the statement "we will share your data with our business partners" in itself is clear, its scope is not. There is often clearly room for improvement.

Consent is understood by many service providers as a necessary requirement for entering into contracts, and for being allowed to collect and use personal data. It is usually implemented, if at all, by means of an "I agree" button. The user has no choice but to accept the privacy conditions set by the service provider if she wants to enter into a contract.

PRIME replaces the 'take it or leave it' approach to privacy policies by a system of policy negotiation. Both parties can express different kinds of policies relating to authorisations, data handling and preferences. The user is assisted (see *human measure*) by the PRIME Console which helps in setting personal preferences and requirements, in converting preferences from machine readable form to human readable form and vice versa, and in automatically negotiating the user's preferences with the other party. It supports the notion of user roles that allow the user to define policy sets (and their associated personal data) for various frequent uses. The PRIME Console therefore allows the user to delegate reaching a policy agreement to a digital assistant for common interactions and assists the user in more complex interactions.

Alice, for instance, has a preference to reduce the chances of receiving unsolicited email. Therefore she wants to receive order confirmation through

---

[6] The enterprise, on the other hand, also wants to have certainty that the customer meets her obligations, such as payment for the goods or services, either directly or through a trusted third party.

a temporary, 'disposable' mail address that retains mail only for one hour. Furthermore, she does not want to receive newsletters, unless the shop offers some kind of incentive after her initial refusal, in which case the PRIME Console has to consult her. She also does not want to have her data distributed to business affiliates.

When the user enters a PRIME enabled website, she can activate the PRIME Console to take over all interactions relating to privacy policies or personal data. User applications may delegate identity management to the PRIME Console which then replaces the traditional webforms by a unified interface to the user's identity management system.

The PRIME Console keeps track of personal data relating to the user, her (negotiated) policies and service customisations, as well as of data disclosure to PRIME enabled services. The Console therefore keeps track of the history of the user's interactions. It can also poll services to provide information about the use of the data (and further disclosure to other parties) by this service provider, as well as the state of policy enforcement because the policies are associated with the data (*sticky policies*). This allows the user to maintain control over her own data and exercise her statutory rights[7] to be informed about the data controller's use of her data in a more effective way.

Data minimisation is furthermore facilitated by support for pseudonyms. In fact, anonymous, or pseudonymous interactions are the default within PRIME. In many cases a handle to the user (or pseudonym) known by both parties is sufficient for the interaction and for possible follow-up interactions. For instance returning customers can be recognised on the basis of the user's pseudonym, and also tailoring services to her needs and preferences is possible on the basis of a pseudonym. PRIME supports different forms of pseudonyms with different characteristics with respect to linkability between the pseudonyms.

Using *pseudonyms* instead of civil identities in transactions makes it more difficult to validate *claims* or attributes.[8] Yet, claims play an important role in minimising data disclosure because often it is not the identity of the user that matters but rather some attribute. For instance, the fact that Alice is over 16 years of age allows her to purchase alcohol, not the fact that she is called Alice. The fact that she can make the warranted claim that payment is assured, such as providing valid, non-revoked, credit card details, should be sufficient reason for CyberWinery to authorise shipment for a box of wine.

Claims in the real world can be certified by third parties. The State, for instance, offer certificates that a certain individual has a certain date of birth and lives at a certain address (passport, ID card, or driver's license). Online certifiers can, by means of cryptographic techniques (security tokens), vouch

---

[7] As laid out in for instance the Data Protection Directive 95/46/EC.

[8] If I know your name, I can try to get data about you through all sorts of channels, which is much more difficult if I only know you by transaction pseudonym ghT57897.

for certain claims in a secure manner that cannot be tampered with. PRIME offers extensive support for certified claims as well as for the creation of *private credentials.* Private credentials (or certificates) allow for releasing partial information contained in a master certificate, for example, that one is over 18 using the birth date attribute. In addition, it is possible, to provide encryptions of attributes of private certificates in the claim together with a proof that the encryptions actually contain the third-party-endorsed attribute values and not any values put there by the claimant. Alice uses such a private certificate to prove that she is over 18.

What data Alice discloses when ordering her box of white wine depends on her preferences. She may want to reveal her real identity to CyberWinery, but she can also opt for a pseudonym. In the latter case the remainder of the shopping process will be slightly more complex than in the traditional setting where providing name, address and credit card data are sufficient to complete the transaction. If the winery makes use of a delivery service there is no need for them to have her address for the purpose of delivery. Alice can provide CyberWinery with a security token that points to her account with the delivery service. Alternatively, she could send an encrypted token including her address to CyberWinery while only providing the delivery service with the decryption key to her address.

1.3.4 Phase 4: After-Sales and Delivery — Retaining Control: Policy Enforcement

Some time after Alice placed her order she is not only curious to know when to expect her purchase, but she is equally eager to know what data CyberWinery actually stored about her. She even had second thoughts about the shop having information about her at all. However, because the PRIME Console created a transaction pseudonym for her, she has trouble remembering which pseudonym was used for the transaction.

This shows two core problems of (data protection in) the online world. The first is that (privacy savvy) netizens will accumulate many digital personae. They use avatars in online games and virtual realities, pseudonyms for other kinds of interactions and finally their civil identity for certain business. Unless there is a way to keep track of what each of these partial identities has done online, privacy protection is difficult in practice. The second problem is the lack of control on information once it has been released. Unlike goods, data cannot be reclaimed without the possibility that a copy is left behind in several possible places. This makes erasing traces hard, unless technology is brought to bear.

PRIME supports the user in staying in control of her partial identities, also after data disclosure. It offers support for managing the (possibly) multiple pseudonyms that make up a partial identity and the revealed (certified) attributes of the user under these pseudonyms. It provides the user with three

central means to accomplish this: tracking one's data trail, support for rights enforcement and policy enforcement.

The PRIME Console's DataTrack function maintains a database of the personal data disclosed by the user. It provides a comprehensive overview of what personal data the user has released to whom, under which partial identity (pseudonym), when, and for what purpose (i.e., under what policy). The DataTrack therefore is an essential tool to keep track of one's digital personae.

The DataTrack also assists the user in enforcing her rights under the Data Protection Directive, for instance the right to get information about the data the service provider has about her, the right to correction and erasure. This functionality requires the implementation of PRIME Middleware at the user's side and the server's side. In cases of non-PRIME compliant service providers, the DataTrack will provide the user with hints on how to correctly enforce her rights using legal means.

The most powerful function of the PRIME concept is the technical *enforcement of agreed policies* on the service's side when equipped with PRIME enabled Middleware. The machine-readable part of the sticky policies can be processed automatically by the PRIME server Middleware. The system will detect the fulfilment of certain conditions that warrant action on the user's data. For instance, it may detect certain purposes of data collection having been fulfilled, e.g., the order was shipped and hence retaining the shipping address is no longer necessary. In line with the principle of data minimisation it will then be deleted. Or, if the user allows the service provider to store her home address for six months for personal offers, the expiry date is attached to the address. The server side PRIME Middleware will then automatically delete the home address at the due date.

Ideally, the user's increased control over the data disclosure should lead to the disclosure of less personal data, but better quality data. As a side effect, certainty over policy enforcement may increase the chances of the data being accurately provided instead of being fabricated. This not only is beneficial for the user, but also for the service provider. Automated policy enforcement is also advantageous for service providers because it facilitates compliance with internal policies as well as legal regulations.

1.3.5   Phase 5: Customer Relationship — Building the Relationship

The quality of the CyberWinery's dry white wine appeals to Alice's taste and she returns to the shop to try out some of their red wines. She becomes a returning customer and before she realises it, she is a frequent customer (being the one with a big house, she hosts many family parties). Alicia's expertise as a wine buff turns out to be limited to white wine, so Alice decides that she may need the shop's recommendations on red and sparkling wines. She might also be interested in getting recommendations based on her previous

purchases, similar to recommendations given at Amazon when accessing the site as a frequent customer. Both CyberWinery and Alice may benefit from this. Provided that Alice consented to such a service, CyberWinery could provide it. The PRIME Console facilitates the means to opt-in and opt-out of such a recommendation service at will.

She may do so if she is concerned about the store's ability to build detailed profiles about her, or even combine their data with those of other service providers to create a comprehensive picture of their customers' tastes, budgets and more. Although CyberWinery's recommendations may benefit from such detailed profiles, Alice wants to remain in control.

This desire to benefit from the advice provided by a service provider who is familiar with one's personal history on the one hand, and to remain relatively unknown on the other, leads to identity management issues. PRIME can help to address these. PRIME allows for a reduction of linkability of personal data if the user adopts different kinds of pseudonyms during the interactions. Alice can enter the store and identify herself with a role-relationship pseudonym for browsing and choosing items at CyberWinery that allows the shop to build a 'shopping' history for this pseudonym that is unlinkable to her real identity. Only when she decides to order, she switches to a transaction pseudonym that is only maintained for this specific transaction and is unlinkable to her role-relationship pseudonym. CyberWinery will retain the data associated to Alice's role-relationship pseudonym for further interactions. This does require a certain infrastructure to be in place that allows for a seamless identity switch at Alice's end – items placed in her shopping basket while browsing under her role-relationship pseudonym should be transferred to the real shopping basket she uses when checking out under her transaction pseudonym. The PRIME Middleware allows for this. CyberWinery also has to be trustworthy not to associate the two pseudonyms behind the screens.

There are other concerns during online interactions. What about Eve the notorious eavesdropper? Alice does not have to worry much about people acquiring her personal through interception of her communication because her personal data will be communicated using keys from the service provider and herself unavailable to Eve (public key encryption). Alice will also have some protection against 'man in the middle attacks', such as spoofed websites, because the PRIME Middleware will help her detect whether the site she visits is false, and again her personal data will be communicated using keys from the genuine site and herself.

1.3.6 Phase 6: Beyond Being a Connoisseur — Alice's Other Identities

It appears Alice has found a new hobby. She begins to like good food, good wine and matching company. She also appears to have a good nose and matching taste. She quickly gains a reputation as a connoisseur which also becomes

apparent in online communities. In one of them, iConnoisseur, she gains a reputation of being a real expert under her pseudonym Malbecky. iConnoisseur's reputation system is based on the member's rating of the amount and quality of others' contributions. Alice receives 6 out of 10 corks in a whim. When she joins CyberWinery's forum, she learns that the quality of discussion is much lower here and she decides to contribute to improve the forum of her favorite webshop. However, as a newcomer she has trouble being heard. If only she could bring in her reputation.

This anecdote illustrates a common problem in the online world. Netizens build up reputations such as financial creditability, but also valuations and ratings by peers, such as iConnoisseur 'corks' are common. Transferring reputations from one context to the next, without linkability of the underlying partial identities is a feature that will prove valuable in online interactions.

PRIME can handle this kind of reputation transfer because reputations can be transferred into (anonymous) credentials. iConnoisseur can provide Alice with a credential that she can present at CyberWinery's forum. CyberWinery can check the validity of the credential, without being able to establish a link to Alice's pseudonym in the iConnoisseur site.

Now that Alice has become a real connoisseur, she starts thinking about a career shift. She visits many vineyards in Spain, Italy, and France. She notices the steep price differences between CyberWinery and local vineyards and sees a business opportunity. She and her bookkeeping genius of a sister Alicia set up a small online wine shop which implements the PRIME Middleware to honour their customers' privacy.

Their shop, MerchantSisters, flourishes, but one of their customers, identified as Bob13, plays a trick on them. He (or she) does not pay for a large shipment after a number of successful transactions. The sisters want to claim payment but need a way to address Bob13 who does not respond to mail sent to the email address he provided.

PRIME allows for several new business mechanisms for privacy-enhanced services. The classical approach would be to use a payment system that adopts the first line responsibility for paying the service provider, which is how current services like credit cards deal with the issue. The problem introduced by Bob13 would not have occurred in this situation, or would have been put on the plate of the credit card company.

But with PRIME and its use of credentials and pseudonyms other approaches become feasible. Anonymity and pseudonymity have their limits. As users and service providers should be accountable for their actions when they breach their contractual or legal obligations, also when they are surfing the web. Users can use pseudonyms and credentials to minimise data disclosure as long as there are mechanisms to reveal their civil identity when warranted, and understrict conditions. One of these conditions would be the use of a trusted third party that is contractually bound to reveal the civil identity of the user under certain circumstances (i.e., breach of contract between the MerchantSisters and Bob13 in our case).

Another approach would go even further and have the trusted third party act as a court of arbitration. The contract between the MerchantSisters and Bob13 could contain a clause subjecting both parties to the rulings of this court. In many cases, alternative dispute resolution can work cheaper and faster than regular courts - also effectively lowering the threshold for making sustained claims. Involving the trusted third party as an intermediary preserves Bob13's privacy if the claims of the MerchantSisters prove to be unsubstantiated.

1.4   The Bigger Picture

The preceding pages have illustrated some of the (privacy) issues that individuals and businesses encounter in online interactions and the ways in which PRIME can offer privacy-enhancing solutions to these problems. The scenario introduced a limited application domain, online shopping. The PRIME concepts can also be used in other application domains, and also in other forms of communication. Here are some examples.

The adoption of mobile phones and other mobile communication equipment is enormous. Because the location of these devices can be determined by telecommunications providers, this opens the way to a plethora of Location Based Services (LBS). One of these developments involves pull services. Here, the user initiates a location determination which is then used to provide a location based service, such as pointing out the nearest train station or pharmacy. Push services are also possible. Here the service is activated without the individual's intervention. The location of the device triggers services the user subscribes to. For example a service could inform the user that one of their friends is nearby. These scenario's are likely to involve multiple service providers: the telecom infrastructure provider, content service providers and telecom providers. It may be undesirable for these different providers to have access to the data generated by location based services. For instance, why should the telecom provider, let alone the infrastructure provider, know that Alice is looking for a pharmacy? PRIME technology can be used in LBS provisioning to offer ways to keep these various service providers separate and thereby maintain the unlinkability of the user's personal data. This scenario is the basis of one of the PRIME application prototypes.

Another important area where PRIME concepts can be of service is in citizen government interactions. Current eGovernment services and identity management infrastructures are not exactly ideal from a privacy perspective. Adoption of PRIME technology in eGovernment would open ways for pseudonymous interactions while also allowing identified interaction, when required. This use runs parallel to Alice's shopping scenario. The added bonus is that the government can serve as a credential provider which would leverage privacy-enhancing technology from beyond eGovernment use to private sector use because there is a clear need for certified credentials here as well.

A third area where privacy issues can be tackled by PRIME technology are social networks. Profile sites, self-help discussion forums, and even virtual communities such as SecondLife are environments where the users are very open about their interests, attitudes, concerns and behaviour. Though this is not without problems. The mechanisms controlling access to personal data are coarse in most cases. For instance, friends, and friends of friends, can have access to your profile data. It becomes increasingly clear that elaborate schemes are necessary to curb the spread of personal data, for instance by distinguishing types of stakeholders: friends, colleagues, sporting mates, etc. PRIME concepts can help here to define circles of users, decide who gets access to what data, offer encrypted data to be unencrypted only by authorised 'friends', and allows the user to see who had access to what data.

1.4.1   Concepts and Human-Computer Interaction

The preceding sections have illustrated some of the PRIME concepts[9] and some possible uses. Introducing and adopting privacy-enhancing identity management not only makes online life possibly easier, for instance by enabling portable identities, it also means that individuals and businesses have to adopt different kinds of concepts and modes of operation. Data minimisation also means a change of attitude and culture. But beside this, relatively novel concepts such as roles, use contexts, credentials, and certificates are required. Although most people (implicitly) use the concept of social roles, for instance Alice is Alicia's sister, entrepreneur, tennis player, and possibly also mother, this use of role concepts to delineate access to personal data will be new to them. Yet these kinds of concepts are prerequisites for more elaborate privacy-enhancing identity management systems.

Privacy-enhancing identity management is not mature but a field in flux and it is still in the research phase. This means that, although the underlying technical mechanisms are relatively clear, the translation of these to concepts understandable for the normal user are not yet completed. In this respect, the user interface to the identity management system plays an important role because it is the user's instrument and shields the user from the technical intricacies. Much work in this field remains to be done on the level of requirements, the conceptual level, and in designing concrete interfaces. Some approaches in this field are also shown in the PRIME project.

1.4.2   Public Awareness

Privacy issues abound, and to some extent solutions are also present. Yet the adoption of privacy-enhancing solutions by businesses and individuals has so far lagged behind what may be necessary to bring the Internet to full fruition.

---

[9] More detailed (technical) information can be found in the PRIME Architecture V2 and PRIME Framework V2 documents.

This is partly due to a lack of awareness among the general public of the risks involved in the unbounded disclosure of personal data. Reports in the popular press about privacy incidents involving personal data leaks from enterprise and government databases, about profiling and mining an individual's past on profile sites by human resource departments and reports about ID theft surface more frequently. This may slowly increase the public's awareness that to be more careful with their personal data than they think. The PRIME project sees it as one of its tasks to raise public awareness with respect to privacy issues in a more systematic way. This book, white papers, but also general public tutorials and promotional videos are part of this work package.

1.4.3   Economics

Businesses are utilizing data, in particular personal data, and so personal data routinely for daily operations, and as means of customising services, e.g., to employees and customers. Some of these information-processing practices are coming under increasing scrutiny leading to a call for better privacy management in organisations. Some processes may even become impossible to execute because of limitations imposed by privacy regulations and policies. In definitional terms a business process is a structured, measured set of activities designed to produce a specified output for a particular (internal or external) customer or market. The central question that concerns PRIME is how business processes are impacted by personal data, and how they can be reengineered to improve their privacy management. Realizing an adequate level of data protection requires the implementation of a set of organizational/procedural, e.g., segregation of duties and data handling procedures and technical measures. The latter are usually described as 'Privacy Enhancing Technologies' (PETs).

For the implementation of PETs solutions and PRIME in general, a increased level of maturity of the organization is often required. It is highly unlikely that an immature organization will implement PETs, let alone that these organizations have any awareness of privacy protection. For privacy in particular we believe that there are two levels: the level where privacy is at best an ad hoc process, with local patches to solve local privacy problems; and the level where privacy is subject to a focused company policy.

The benefits offered by PETs can be quantitative or qualitative. If the application of PET leads to a reduction in costs or increase in revenues (e.g through a bigger market share), then the benefits can be measured and, therefore, are quantitative. Qualitative benefits are tricky to measure and hard to express in monetary terms; however, they can surpass the quantitative benefits. One example is the positive image generated by the application of PETs.

Costs of PETs vary with the selected PETs option. For example if the option is data anonymization the emphasis lies on the one-off investments and less on the structural costs. When data are separated, different domains are created, the data model usually has to be modified, and there is more

often a need for customization to implement the PET option. Encryption, for instance, is often cheaper than the application of biometrics with PKI.

1.4.4   Reaching Out

Finally, in order for privacy-enhancing identity management to be adopted on a large scale not only requires that individuals take notice of the technology. But it also requires service providers to implement the necessary software. Businesses and governments will only do so if they see an advantage for doing this. PRIME investigates and reports on business opportunities, costs and benefits in order to show the viability of adopting privacy-enhancing identity management. It allows businesses and governments, for instance, to comply with data protection legislation more easily. It may also reduce their liability because storing less personal data means less vulnerability to attacks by ID thieves. Not asking for excessive data and offering ways for pseudonymous transactions may also increase the quality of the data they have about their customers.

Another prerequisite for large scale adoption is interoperability. PRIME, or for that matter any identity management system, stands no chance unless it allows interoperability with existing back-end applications and other identity management systems. This calls for standardisation. The PRIME project is therefore actively involved with standardisation bodies, such as W3C and the relevant ISO/IEC Working Groups.

1.5   Requirements for Identity Management Systems

At the start of the PRIME project in 2004, the following principles were adopted as guidelines for the design and implementation of privacy-enhancing identity management solutions:

- Design must start from maximum privacy;
- Explicit privacy governs system usage;
- Privacy rules must be enforced, not just stated;
- Privacy enforcement must be trustworthy;
- Users need easy and intuitive abstractions of privacy;
- Privacy needs an integrated approach; and
- Privacy must be integrated with applications.

The PRIME project continues to adhere to these principles. In the PRIME white papers we have approached requirements for privacy-enhancing identity management from a slightly different angle and have combined the PRIME principles with requirements brought forward by other initiatives. This has resulted in the following list of requirements:

User control and consent. In order to maintain the individuals' trust in the information society and guarantee their freedom of choice (autonomy), users should be able to control which personal data are given to whom and for what purpose. Exercising control requires informed and uncoerced consent for specific uses, which may be revoked at a later date, by the individual.

Justifiable parties. Personal data should only be accessible to entities with a legitimate interest in the data, e.g., by consent of the individual, by legal obligation or for other legitimate purposes. Service providers should implement technical measures to enforce this requirement, especially with respect to the use of personal data by third parties (for secondary uses). This requirement also implies that the user should be able to check the authenticity of the data requester.

Data minimisation. Personal data disclosure should be limited to adequate, relevant and non-excessive data. Implied in this requirement is that data needs to be provided on a need-to-know basis and stored on a need-to-retain basis. This requires the requester to specify the purposes of collection, processing and storing of the data. Data should be deleted at the requester's end as son as the specified purposes of data collection are met.

Policies and policy enforcement. Users should be able to express their privacy policies and preferences and negotiate the terms of data disclosure with service providers. The agreed upon policies should be strongly enforced by the identity management systems on both sides of the transaction.

Human measure. The user should be able to understand how she can exercise control over her personal data. Communication should therefore be in plain language using understandable concepts. 'Thingification' should be used for necessary but complex notions, such as roles, rights and obligations (e.g., using business cards to represent data related to a role). Human-machine communication within and between contexts should be unambiguous offering situational normality and predictability. The interface should help to protect the user against identity attacks.

Multiple identities and accountability. The user should be able to use a range of identifiers with varying degrees of observability and linkability. This means users must have a choice to operate anonymously, pseudonymously or known. Users should also be able to use identities provided by public bodies or enterprises, as well as ones created by themselves, to be able to provide certainty about their identity to other entities and therefore promote accountability when required.

References

[Art04]   Article 29 Data Protection Working Party. Opinion on More Harmonised Information Provisions, November 25 (2004),
http://www.europa.eu.int/comm/justice_home/fsj/
privacy/docs/wpdocs/2004/wp100_en.pdf

[Ass]   Canadian Standards Association. The canadian standards association's privacy code, http://www.csa.ca/standards/privacy/code/

[Cam05]   Cameron, K.: The laws of identity. Technical report, Microsoft Corporation (2005)

[Cav06]   Cavoukian, A.: 7 laws of identity, the case for privacy-embedded laws of identity in the digital age, p. 18 (2006)

[Cou81]   Council of europe: Convention for the protection of individuals with regard to automatic processing of personal data, Strasbourg, January 28 (1981)

[LAP06]   Liberty Alliance Project. Whitepaper: Personal Identity, March 23 (2006), http://www.projectliberty.org/liberty/content/
download/395/2744/file/Personal_Identity.pdf

[Org80]   Organization for Economic Co-operation and Development. OECD guidelines on the protection of privacy and transborder flows of personal data (1980), http://www.oecd.org/document/18/
0,2340,en_2649_34255_1815186_119820_1_1_1,00.html

Part II

Setting the Stage

2

Overview and Introduction Part II

Ronald Leenes

Tilburg University

2.1 Introduction

The internet reached the general public in the early 1990s. Since then it has changed dramatically. In its early days it was primarily an information source where its novel users could marvel about what new ways of information dissemination, such as Gopher and later the World Wide Web had to offer. People also communicated. For instance by means of email, which typically involves communication between people who already know each other or are aware of each other's email address. Other types of communication involved Newsgroups, bulletin boards and IRC channels where its participants did not have to know each other in advance.

Gradually, the internet changed into an infrastructure where everyone participates and where two way interaction for many is everyday practice. This change has brought one of the design omissions of the internet to light. The internet was not fitted with an identity management layer or mechanism. The internet does have an ID infrastructure, but this is based on identifying machines, not humans.[1] The effects of this lack of a proper ID infrastructure are becoming more prominent every day. It turns out that on the one hand we need mechanisms to identify or at least recognize (returning) people online more often than in the offline world, yet on the other hand there is also a need not to be identified or recognized online. This evidently creates tensions that need to be resolved.

In the physical world most people can go about relatively anonymously and unobserved. The local bakery in my small community may know its customers,

---

[1] One can even question whether the ID infrastructure identifies machines properly, given the ease by which phishing and spoofing can be set up.

but this is already no longer the case on a slightly larger scale. Initially, the internet seemed to offer its users the same or an even greater sense of anonymity than real life. Steiner's (1993) now famous cartoon in the New Yorker, depicting a dog at a computer screen remarking to another dog that "On the Internet no one knows you're a dog", aptly reflects this idea. Some 15 years later, we know better, but the signs where already there in 2000 when Tom Toles, the cartoonist for the Buffalo News, revisits the dog scene. Following Steiner, he pictures two dogs marveling at this invention called 'the internet' in front of a computer screen. One of Toles' dogs, reminiscent of Steiner's clever dog, remarks to the other, "The best thing about the internet is, they don't know you're a dog." Anonymity is apparently not only a feature of this novel network, but indeed, *the* feature from his (or is it her?) perspective. The second part of the cartoon clearly shows the dog's ignorance regarding the internet's true nature. It shows the two dogs watching the screen which painfully faults the protagonist's belief by displaying "You're a four-year-old German Shepherd-Schnauzer mix, likes to shop for rawhide chews, 213 visits to the Lassie website, chatroom conversation 8-29-99 said third Lassie was the hottest, downloaded photos of third Lassie 10-12-99, e-mailed them to five other dogs whose identities are ....".

I am certainly not the first to use the two cartoons to illustrate anonymity and privacy (or the lack thereof) and the internet (e.g., [And05, GKM07]). Yet, I want to draw the reader's attention to some aspects of Toles' cartoon that are not entirely apparent on first inspection. Toles' cartoon appears to show that not anonymity, but a state of being known, is the current norm on the internet. We are moving in that direction, although Toles exaggerated — our online behaviour is not that transparent —, it certainly was not in 2000.

Next to this first obvious observation, the cartoon also shows us the other relevant characteristics of the internet. For instance, the protagonist is not addressed by name. Whoever or whatever is responsible for displaying the information on the computer screen apparently has detailed information about the dog's features and even behaviour but, judging from the message, does not know or display the dog's name. This is interesting, because names are common identifiers and knowing someone's name is associated with knowing the person. The image forcefully shows that names are sometimes unnecessary in characterising or identifying individuals. By revealing intimate details, the sender conveys that he knows the subject and we as readers recognise that the intimate details identify the dog, even though we don't know the dog; the subject's name is irrelevant. The image thus hints at a salient feature of modern profiling. Not so much traditional identifiers, names and addresses, matter, but rather, the capability to recognise a particular individual and being able to associate the (inferred) features and behaviour that are represented in profiles to this identifiable individual.

A second implicit message in the cartoon is a reference to how these profiles come about. The dogs are taken by surprise. The protagonist assumes that (s)he can go about anonymously on the internet, but the opposite is

the case. The collection of (intimate) information that was displayed on the screen occurred without the subject's awareness and, as we may assume, their consent. The collection of personal data online is conducted in an opaque and unobtrusive fashion, yet its results are striking; the observer really 'knows' the cartoon's protagonist. This again hints at profiling. Search engine providers, web publishers and Internet Service Providers (ISPs) alike collect the data traces that are left by internet users during their daily affairs online. These traces are stored, combined, exchanged and accumulated into potentially detailed profiles of these individuals. Beyond an abstract awareness that this happens, most users have no idea what specific data is collected by whom and for what purpose.

A third message in the picture is that we should mind that these profiles exist. This message is less explicit. Steiner's dogs celebrated the freedom and emancipation that are offered by the internet. Toles' joke hinged on the realisation that the inverse is the case; the internet, in a sense, curbs the individual. The individual's identity is known and they are being observed. This affects their freedom to act and, therefore, the picture shows a state of affairs that should make us think about its desirability.

The PRIME project has taken up this challenge, and so have others. The need for online identity management is acknowledged by both enterprises/governments and customers/citizens. In recent years a plethora of identity management initiatives has surfaced. Each of these initiatives aims to resolve particular issues. Some focus on improving access control for enterprises and aim at implementing large scale enterprise centric solutions. Others, typically aiming at end users, try to help the individual in keeping track of their usernames and passwords. Most initiatives and projects resolve partial issues.

What is needed is a holistic approach to develop comprehensive solutions that technically enforce strong privacy, are based on the European regulatory and legal framework, and are socially acceptable and desirable, economically exploitable, intuitive and user-friendly, deployable by applications. Part II of the PRIME book discusses these issues in more detail and derives a set of requirements that provide the foundation for the PRIME technology.

2.2   An Approach from Three Perspectives

Individuals engage in different social and economic relations online. How they present themselves is partly determined by themselves and how they want the world to see them, and partly determined by others who 'demand' to see certain aspects of the individual. What information is provided in the various different relations is diverse. In practice, the result of all these interactions is that each individual explicitly of implicitly creates many online partial identities or digital personae ([Cla94]) over time. People want and need to be able to keep these different personae confined to their specific contexts (one aspect of privacy). Identity management is therefore a social need and insight

in the nature of this social need is required in order to understand what functions for the individual should be supported by identity management systems.

The social needs regarding privacy and identity management have found their way into regulation, especially in Europe. The European legal framework is based on a set of principles that convey the European privacy values. Understanding these principles is important in making balanced decisions regarding the various interests at stake in the identity management landscape. The legal framework also provides a set of legal requirements that have to be taken into account in developing identity management solutions.

The third perspective that is required to properly understand the need for privacy enhanced identity management is the business perspective. Identity management provides the interface between the individual and the enterprise/government. Understanding the company perspective is therefore a prerequisite for identity management development.

2.3   Structure Part II

This Part of the book starts with an overview of the identity management landscape (Chapter 3). It introduces two different perspectives on identity management, an enterprise view and an individual view. The enterprise view concentrates on access control (Identification, Authentication and Autorisation) to resources that is usually implemented as a system of user accounts. The individual perspective is based on the way individuals manage their identity in everyday life. Identity in this view relates to the way individuals present themselves to others and how others view them. The chapter further discusses Identity management developments from an enterprise perspective via identity federation towards user-centric IdM. The chapter concludes with an overview of developments that further complicate the identity landscape: web 2.0, the Internet of Things.

Chapter 4 discusses the need to incorporate privacy into identity management systems. It starts by discussing that there is an individual interest in privacy protection online in general and in IdM more specifically. It then moves on to argue that also from an organisational perspective, the domain of enterprises and governments there are clear indicators that privacy needs to play a more important role in IdM. The third level discussed in this chapter is the societal level. Here it is argued that privacy is a common, public, and collective value that benefits society as a whole. Europeans share a common understanding that privacy matters even though we may disagree to what extent. This warrants treating privacy as a common good. Privacy also resembles a public good such as clean air: we all benefit from its existence and when it is constrained not only individuals but society as a whole will be harmed. Privacy is also comparable to collective goods in the sense that guaranteeing and enforcing privacy on the individual level does not really work.

Chapter 5 discusses the existing legal framework regarding privacy and data protection. The chapter starts by a brief introduction on the European history of data protection regulation. Next it describes the core principles of the EU data protection regulation and derives a set of concrete design requirements from these principles. The chapter then focuses on some of the issues regarding the applicability of the current legal framework in an evolving online world. The protection seemed adequate at the time the Directive was written. The tide, however, seems to shift. The development of new technologies and new services create new challenges with respect to privacy and data protection.

Chapter 6 handles the common legal-social requirements for privacy-enhanced identity management systems. It starts with discussing the importance of audience segregation in Identity Management, and its direct link with privacy. Audience segregation is then further elaborated in the guise of user control. User control is decomposed into a set of requirements that capture legal and sociological/psychological needs. The chapter is concluded by discussing a number of adoption requirements that should guarantee user adoption of privacy-enhanced identity management developed along the lines of the previous requirements.

Chapter 7 focuses on the business perspective of identity management. The basic question explored concerns economic motivations for an organization to invest in privacy and identity management. The analysis starts with a discussion of technology adoption processes. Next a maturity model regarding identity management processes is described that suggests that we may only expect enterprises that are sufficiently mature regarding their identity management and that are sufficiently privacy aware to be able and willing to implement advanced privacy-enhancing technologies. The chapter then discusses a cost/benefit analysis model for investments in PET implementation. Privacy protection is currently seen as a negative cost driver in a cost/benefit analysis. Finally a set of business requirements is introduced.

3

The Identity Landscape

Bart Priem[1], Ronald Leenes[1], Eleni Kosta[2], and Aleksandra Kuczerawy[2]

[1] Tilburg University
[2] KU Leuven

3.1 Introduction

Many people will have an image of 'who they are' and how their identity is established. Moreover, most individuals will probably relate the concept of identity (and identity management) to their reputation as an individual, how they define themselves, and how others look at them. In this view, identity relates to the personal aspect of identity. However, the term identity is also used in many other ways, for instance in the sense of cultural identity — what makes an Englishman English? —, or in the sense of identity management in IT systems. Because of this, a clear definition of 'identity' is difficult to provide.

One of the developments that influences the notion and the use of the term identity is the development and use of Information and Communication Technologies (ICTs). Especially the creation of the 'online environment' has added complexity to the notion of identity. The online environment, for instance, lacks a clear ID infrastructure. It was designed to identify the endpoints of communication, which typically are devices (such as computers) that are, or were, shared by multiple individuals. Important aspects of identification in the offline world, such as the presence of the body as a means to recognize and identify individuals is lacking online. Instead, internet-facilitated interaction currently relies heavily on information that can be manipulated and that has unclear status to identify and represent human beings and devices. Because of this, several initiatives exist to improve online identity management (IdM). All these initiatives operate in a rapidly evolving field with moving targets and changing issues. Furthermore they need to deal with the diverse interests of the various stakeholders.

To put the PRIME project, its technology, and its vision in perspective, this chapter will provide an introduction to the current landscape of online identity and online identity management. We will discuss some of the meanings of the term 'identity' and describe developments in the identity management field which can be summarized as an evolution from enterprise centric towards user centric solutions. We will conclude the chapter with some complicating developments that illustrates issues to come that need to be incorporated in any comprehensive identity management system. This chapter serves as a foundation for the chapters to come. It does not, however, provide an extensive overview of the philosophical and sociological aspects of identity.

3.2 The Concept of (Online) Identity

Identity is a dynamic and contextual concept. It has several dimensions. It is, for instance, used to represent a person, but is also used to identify and recognise such a person. Thus, identity is used both in descriptive terms and process terms [WP205]. One can furthermore refer to identity as to who a person 'really is' (sometimes called 'ipse identity'), but also as to how a person is characterised or represented by himself or by others (or 'idem identity'). There is thus a difference in the notion of identity from a philosophical point of view (who someone really is) which regards identity as fluid and indeterminate, and the more 'practical' view on identity which relates to the static representation of an individual in a certain context in the form of a set of attributes related to this individual (see [WP205]).

When identity is considered in the context of online identity management, we mainly deal with the static identity of an individual (represented in data) and its composition and deployment throughout online contexts. In the online environment, identity management primarily relates to the composition of an identity out of 'identity information' that relates to an individual or another entity that acts in this environment. In this sense, both human beings and devices can have an online identity; historically, device identity preceded human identity in the online environment because the internet was developed as a computer-to-computer infrastructure [Coy07, Cam05].

Both online and offline, individuals interact with people and organisations in many different relations. All these relations concern the exchange of information and/or attribute-value pairs. Different (kinds of) relationships involve different parcels of information and therefore individuals present different images of themselves in different contexts. A single individual therefore consists of different characterisations tied to the different contexts in which she operates. For example, the co-workers in a work-related context will characterise an individual differently than the friends that interact with the same individual in the context of friendship. The relevant attributes associated to an individual are different in a working environment than in a social environment and individuals may also represent themselves differently throughout

such contexts. As we will see later in more detail, this capability to keep the different contexts separated, 'audience segregation' [Gof59], is an essential characteristic of modern (western) societies which allows for different kinds of social relationships to be established and maintained [Rac75].

In the online environment, the different manifestations of an individual can be defined as partial identities, or 'digital personae' [Cla94], which are constructed from the information people give, or 'give off' in a relation [Gof59]. The construction of a partial identity is not solely based on information that is determined and controlled by the individual to whom an identity relates ('projected' in Clarke's terms). Others, the recipients, may construct their own image of the individual by observing them or their behaviour as represented in data and they may add information to an existing partial identity (which leads to 'imposed personae' in Clarke's terms). The information contained in a partial (imposed) persona may not always be known to the individual concerned.

Partial identities in the online world are thus determined both by information known and unknown to the represented individual and this information may be controllable or uncontrollable by the individual. Moreover, the perception of a partial identity can be different between the individual to whom an identity relates and the person or organisation that uses such an identity [WP205].

Identity already used to be a complex concept for the offline environment, but in the online world it is even a more 'muddled thing' [Cha06], because the internet provides the possibility of disembodied use of identities (ie. without the individual's bodily presence) and facilitates the decontextualisation and transfer of identities (and identity data). On the internet, traditional 'trust tokens' (e.g., clothing, buildings, driving licenses) are largely absent.

3.3   Asymmetric Perspectives

The field of identity management has many stakeholders with different, and potentially conflicting, interests in the design and use of identity management systems. Consumers, regulators, and enterprises can have different perspectives on the concepts of identity, identity management, the online environment, and the use of identity information. 'One-size-fits all' solutions may therefore be difficult to develop and designers need to balance difference perspectives, interests, and requirements. In order to understand these different interests and conceptions of identity and identity management, we will first discuss identity management from the perspective of two principal stakeholders, enterprises (and government) and the individual.

3.3.1   The Enterprise-Centric View on Identity Management

Enterprises and governments have driven the development of identity management systems as a means to know with whom they communicate [OMS+07].

Access control to resources and hence, identification, authentication and authorisation are therefore the key concepts in contemporary identity management. Private *enterprises* that are active in the online environment, make use of identities (e.g., user accounts) to meet strategic objectives, such as ensuring the accuracy of identity information, utilizing the possibilities to store and manage large amounts of data, and the use of information to develop and distribute products and services effectively and efficiently (in a better way than competitors do), and reducing the risk of data loss. The *government* is another major stakeholder. The government needs identity management to provide efficient personalized electronic public services and to prevent citizens from falling victim to fraud and insecurity whilst providing these services. Moreover, the government is a stakeholder in IdM development in general, because IdM promotes the free flow of information in society which can increase welfare, for example.

Identity management developments have been driven by an enterprise-centric view on IdM. Many of the developments that will be described later on in this chapter depart from a perspective that the core function of an IdM system is to manage who has access to certain resources. Online IdM in this view comprises the use of partial identities for identification, authorisation and authentication of individuals to provide them certain services. Central to this kind of identity management are user accounts. These accounts also contain (or link to) data that provides insight in (customer) preferences, purchasing history, and contact data, for example. This information allows the enterprise to create personalised, and customer-oriented services. Most organisations active on the internet keep track of users' purchases, and there is an active market for such customer data [EI06, Tay02].

Enterprise-centric IdM systems focus on facilitating service delivery to the right person, which is 'their' customer or client. The fact that these customers also have accounts at other enterprises which causes inconveniences for these individuals is not a primary concern of the respective enterprises.

3.3.2   A User-Centric View on Identity Management

Individuals are right in the middle of online identity management, because it concerns the management of their identities, and because decisions are made on the basis of these identities. From an individual's point of view, the concept of identity management therefore not only relates to the access control regarding resources. It also, or maybe even rather, relates to how they are manifested and represented, and how this is aligned to their own perception of their identity. Identity management in this sense strongly relates to role playing and presentation of self. The individual should be able to act as an autonomous individual, be able to control their reputation, and have insight in the way they are judged by others in a specific context.

The online environment facilitates the construction and maintenance of projected and imposed personae. Data can easily be collected and combined

into rich personae, transcending the context in which individual bits of information were disclosed. The decontextualisation and combination of data from different sources makes it difficult for individuals to control their different digital personae. This undermines their capabilities to control the image they present in different contexts and to segregate audiences online. The need to do so exists online just as it does offline. People engage in different kinds of activities online (e.g., public, commercial, and intimate) and need to be able to construct matching identities that meet the behavioural rules and requirements set by these different environments.

Important values such as reputation, dignity, autonomy, judgement, and choice are closely related to the individual perspective on identity management. When people cannot determine or control their identity, they may become overexposed, confused, or discriminated, for example. Human beings have an interest in naming and sorting themselves [Gan93, Raa05] and to play different roles. Sometimes they may even need to be anonymous and unidentified (e.g., for purposes of emotional release, relaxation, unpunished criticism, and making mistakes). Individuals appreciate to have a diverse and autonomous life, and need to be able to adapt their identities to the environment they engage in. Even though identity management is not usually the primary goal of the individual, which may explain why many people are not eager to invest time and money in IdM systems [DD08], the social values outlined previously warrant the individual perspective to be taken into account in the development of IdM systems.

3.3.3   Combining the Perspectives

Integrating the different interests in online IdM increases its complexity. There is a clear gap between the enterprise-centric emphasis on customer-relations and access, and the user-side approach which, for instance, requires users to be able to choose different partial identities for different purposes — even within the same system — or be able to use the same partial identity in different contexts [Pfi03]. This gap needs to be closed.

It is also difficult to implement the 'personal' perspective on identity in IdM systems because of the business and government requirements of facilitating trustworthy interaction between them and their users/citizens. We need to acknowledge that the processing of some identity related information is part of the online environment and may be considered necessary in several circumstances. To completely renounce the need for the collection and processing of identity information (personal data) would severely hamper the adoption of such a system by enterprises.

A further complication of integrating both views lies in the fact that multiple parties need to subscribe to the model. Individuals can only use the same or different identities in different occasions and for different purposes if the identity system allows for this, and this requires standards and interoperability.

The fact that the enterprise-centric view to identity management is too limited seems to be acknowledged throughout the industry, as online IdM systems are evolving towards federated systems and recent developments even point towards the development of 'user-centric IdM systems' (coined 'Identity 2.0' by some), which will be demonstrated in the following section. The PRIME-project aims to be at the forefront of these developments.

3.4   Evolving Identity Management Systems

Different models for online identity management have been developed in recent history. Traditionally, identities were managed in so-called corporate identity 'silos'. In this model one single identity management environment is operated by a single service for a specific group of users [Pat03]. Hence, every (online) service had its own identity management system built to their own requirements for authorisation and identification of individuals. From the perspective of users of multiple systems this means that they have to maintain an identity (account) for each and every service they use, which in practice means several sets of passwords and usernames. The 'silo-model' is still a dominant model for identity management on the internet. An obvious drawback of this scheme from the perspective of the users is that it requires them to provide the same (personal) information for every new online service.

The construction of identities in these systems is guided by rules (implicitly) set by the provider of the service. Each account is identified by an identifier. Sometimes these identifiers can be freely chosen, sometimes they have to satisfy certain rules (e.g., at least one number, 8 characters long), or be a valid email address. Individuals are therefore sometimes forced to create different identities (or rather the identifiers that identify the identity) even when they want to use the same identity across domains. Or, in the case of being obliged to use a valid email address, they may have to use identities they don't want to use for a particular use. As a result of these practices two effects on identity construction are visible: one, difficult to remember identifiers as a result of the rules on identifiers imposed by the service provider, and two a convergence of identities to a limited set of partial identities as a result of the requirement to use email addresses as 'usernames'. Furthermore, the 'silo'-approach has resulted in many identity 'one-offs' and an ad-hoc nature of internet identity even though the identities in these silos can be managed by, for instance, storing passwords and usernames in software (password-managers) on a local computer or on a server [OMS+07, Cam05].

A next step in the development of IdM systems has been the development of single organisation single sign-on (SOSSO)[OMS+07]. Here individuals gain access to different resources (applications, web sites) within a single entity's domain once they are authenticated. This kind of IdM slightly alleviates the individual's burden of having to cope with potentially different identities within such a domain. Usually it also limits the individual's capabilities to use different identities within a certain domain (e.g., the association of an account to an

email address limits the number of accounts an individual can establish without also obtaining new email addresses). Effects of SOSSO are the collapse of different (social) contexts within a given domain controlled by the enterprise and linkability because the IdM provider can recognize the individual access to the various resources. SOSSO makes coping with enterprise centric IdM easier for the individual within a particular domain (e.g., company), but does not help when multiple domains are involved.

Multi-organisation single sign-on (e.g., Microsoft .Net Passport) aims to solve this problem, as well as lessen the burden of implementing and maintaining IdM systems within each enterprise in a federation [OMS+07]. In this model, authentication is outsourced to a trusted identity provider (IdP). The IdP identifies and authenticates the user and provides a credential that can be used to access resources from associated service providers. Drawbacks of this model are that the IdP stores the user's data which creates security vulnerabilities. Furthermore, the attendance of one single IdP in all interactions on the Internet creates linkability because the IdP can trace the user after authentication. It also creates a vulnerability (and convenience) because relying enterprises depend on a single IdP involved in all transactions.

Enterprise centric federated identity management (e.g., Liberty Alliance) addresses the problems related to the dependence on a single IdP in a federation, by allowing any number of IdPs to handle authentication. The user authenticates with any of the IdPs in the federation and subsequently can access resources at each of the entities in the federation (where the user has proper authorisations). Some federation schemes not only handle authentication, but also allow the transfer of attributes between the federates [OMS+07]. Federated identity schemes again limit the burden for individuals of having to cope with multiple identities when they want to use a single identity, but do not address the needs of individuals when they want to use different identities for different activities in the federation. The advantages mainly benefit the enterprises which can achieve costs savings arising from a shared scheme based on a standardised, interoperable architecture, and the outsourcing of authentication and IdM to professional identity providers.

Various initiatives in the landscape of 'federated' identity management can be pointed out. Many of these are 'token' based, whilst some are 'anonymous-credential-based systems' (see PRIME's Framework [PRI08]). The traditional token-based systems rely on identity providers that mediate the transactions. The identity provider distributes tokens to the service providers with which an individual interacts. In a token-based system, the service providers still are relying parties (Rp) with regard to the identity attributes they receive. They depend on the IdP, even though some of their vulnerability can be circumvented by means of contracts.

In recent years, a shift from an enterprise centric view to a user-centric view can be observed. Notions, such as 'Identity 2.0' (Sxip, Microsoft Cardspace, Higgins, PRIME, etc) belong in this sphere. In these initiatives the IdP is no longer in the centre of issuing and creating identities, but rather the user is.

In user-centric identity management, the individual's interests are acknowledged in the sense that they manage their own personal data and obtain credentials from identity providers which they can use in their interaction with service providers. Systems based on anonymous credentials even give the user and relying party the opportunity to use identity attributes without the use of a central identity provider [PRI08]. Such systems make it possible to really put the user in the middle of IdM, and thus indicate a shift from an enterprise-centric perspective to a user-centric perspective. The user-centric model provides the user more control over the way they present themselves to others. If designed properly, they assure the necessary level of *privacy* in the online environment.

Federated IdM systems increase convenience for the user to make use of several different services and make identities portable. Furthermore, they can create opportunities for organisations to ease the process of registration, authentication, and authorisation. In addition, these systems allow for cost saving on the retention and collection of data and can create new business opportunities (see [OMS+07]).

3.5   Existing Identity Management Applications

Multiple competing identity management initiatives have emerged in recent years to deal with the Internet's lacking identity layer. These initiatives range from the aforementioned 'identity silos' and 'enterprise centric SSO systems' to 'federated IdM systems'. We will briefly describe some prominent IdM systems.

3.5.1   Microsoft Passport

One of the early initiatives for a cross service identity management is 'Microsoft Passport' (1999). It featured hundreds of millions of accounts due to the fact that Microsoft used Passport for its MSN and Hotmail services. Passport provides the user the benefit of an SSO-experience, and aims to reduce the time a user needs to register and authenticate for different services on the internet associated to Microsoft by means of contractual agreements [OMS+07, PM03].

Microsoft Passport is a web-based service redirecting the user's browser for the purpose of authentication to a central authenticating server. It makes use of Cookies for maintaining (session) credentials [PM03].

In Passport, personal information is stored in a central location (under Microsoft's control) and therefore websites that participate in the initiative rely on Passport for the authentication of users instead of arranging their own authentication schemes [Opp04]. Individuals register at Passport through

Passport's home page, the Microsoft Windows operating system, or via a Hotmail e-mailaccount.

Passport's centralised model makes it vulnerable to attacks and failure. Also, because the system hardly imposes restrictions on user-selected passwords, many users pick easy to guess passwords which increases vulnerability [Opp04]. Furthermore, Passport is based on a single identity provider (Microsoft) which means that it is involved in customer relations of many other organisations. With 'Microsoft in-the-middle', (potential) users and privacy advocates have voiced concerns that this powerful IdP may acquire significant amounts of data about internet activities of the systems users and organisations that make use of the Passport system [Cam05].

Even though Passport provides a simple solution for identity management, it does not fully comply with user requirements and organisational IdM requirements. Especially the dependence on a single identity provider, Microsoft, seems to have obstructed the adoption of Passport in non-Microsoft services. Microsoft's stake in the centralised Passport system has been considered 'out of context' [Cav06]. Another aspect of a centralised IdM system like Passport that could have negatively affected adoption is that it raises concerns in the fields of control over private information, security, and competition [Cho06].

3.5.2   Liberty Alliance

A more decentralised identity management system is being developed by the Liberty Alliance project. This project was initiated in 2001 and has over 150 members, active in education, government, and including technology vendors, as well as many others. The Liberty Alliance aims to develop a federated identity management system with multiple identity providers. Because of this, identity data does not have to be stored at a central organisation whilst users can still have a web based, SSO-experience.

The goal of Liberty Alliance is to establish an open standard for federated identity management. Its technology makes it possible to form 'circles of trust' between trusted authentication service providers (ASP's) and service providers (SP's) [PM03]. Thus, organisations can make agreements with regard to the authentication of individuals and can provide individuals the possibility to use a specific identity within these circles of trust. This reduces the burden for individuals to cope with different identities within certain contexts. For enterprises, the benefit of Liberty Alliance are cost savings from sharing a standardised and interoperable architecture, and from outsourcing activities to identity providers.

Liberty Alliance, however, still relies on organisations that act as identity providers. It focuses on a business-to-business scenario [Pfi03]. Individuals therefore still need to be aware of linkability risks and need to be cautious when they choose privacy policies [PM03].

3.5.3   OpenID

OpenID is a decentralised SSO system, which chiefly aims at lessening the user's problem of having a multitude of passwords and usernames. OpenID-enabled websites relieve the burden for users to remember different usernames and passwords by only requiring them to register at an OpenID identity provider. The advantage of this is that people do not necessarily need to 'sign up' and 'log-in' for every single service on the internet within one browsing-session, but instead can go from one of the sites in the federation to the next once logged in. OpenID rises primarily out of the blogging community but currently both the amount of users and the number of places where OpenID identities can be used is growing rapidly [PR07].

OpenID works with an URL, owned and provided by the individual, that is used for authentication. Websites that require authentication can request the OpenID URL from the individual. The presenter of the OpenID URL is then authenticated by verifying the URL at the OpenID-URL issuer (the IdP). If the issuer certifies that the user actually belongs to the URL, authentication is complete.

OpenID makes use of credentials which are not stored at one single organisation or server. The users can decide for themselves whom they trust with their credentials. Several different OpenID providers already exist, also due to the ease of implementation of OpenID. In addition, OpenID provides a single individual the choice to develop and maintain several different identities at different OpenID providers. OpenID is therefore in the user-centric corner even though users still need to rely on some identity providers.

The OpenID authentication process depends on the redirection of a user to the identity provider's site. This process of redirection raises concerns with regard to 'phishing' attacks (described later in this chapter), because trusted sites can easily be imitated, resulting in a possible exposure of credentials and login information to distrusted parties. This is especially the case when a username and password are being used to login at the IdP's website. Furthermore individuals are still vulnerable to potential unlawful actions of identity providers that can store, collect, and link their data. Moreover, the real separation of contexts still depends on the creation of different accounts, at several servers, requirering extra effort from the individual. For many services on the internet, OpenID is a feasible solution, but some of its design aspects still make it difficult to apply, especially when it concerns 'sensitive' contexts.

3.5.4   Microsoft Cardspace

Microsoft Cardspace is an identity metasystem developed by Microsoft. It is incorporated in Microsoft's operating system Vista. The system uses the metaphor of 'information cards' for the representation of digital identities to provide the individual with a consistent and comprehensible user experience

[Cha06]. Users can create the information cards they want to use by themselves, but it is also possible to use information cards that are issued by third parties like banks, insurance companies, or government.

The Cardspace system claims to circumvent the widespread problem of 'phishing' that occurs when traditional, easily imitated, password-based, web login screens are being used. Microsoft Cardspace addresses the issue of phishing-attacks by 'taking over the screen' of the operating system. Cardspace manages identities at the end user's machine [Mal06]. Moreover, it is an identity metasystem, which makes it complementary to existing identity architectures, like the aforementioned OpenID system. In addition, Cardspace allows users to have different digital identities, regardless of the kinds of security tokens used by other systems. It is therefore also an 'agnostic' IDM system [Cha06].

The user of Microsoft Cardspace is positioned between the relying parties and the identity providers because the information cards are stored in the user's application, which can pass on the information cards to the relying parties when the user chooses. Thus, instead of having one or several organisations 'in the middle', Cardspace facilitates that the user is in the middle of issuing and constructing identities.

3.5.5   Other IdM Systems

There are many other IdM systems under development, for instance, *Higgins*, *Shibboleth*, *Bandit*, *WS-federation*, *Skip* and *Kerberos*. The current brief overview of some of the leading systems suffices for the purpose of this chapter.

It is clear that there is no lack of competition in the identity management landscape [CMBG$^+$02]. The individual perspective until recently has received limited attention though. The same conclusion applies to the privacy aspects of identity management systems. Before turning our attention to these aspects in the following chapters, we briefly review some of the factors complicating the identity management landscape.

3.6   Complicating the Online Identity Landscape

The online environment in which individuals interact and maintain their identities is evolving. From a unidirectional source of information, the internet has become a realm in which many people interact with each other, businesses and the government. Enterprises and governments offer personalized services that require users to establish and maintain online identities. People also increasingly use the internet to maintain their social networks, to relax, to play, or to seek relieve. All these developments have an impact on how identities are constructed and used online and affect the risks that people and organisations take when they are online. In this paragraph we will describe some developments that emphasize the need for IdM systems in which both the personal and the organisational perspective on identity are represented.

3.6.1   The Internet as a Social Environment

The Internet is transforming into Web 2.0 [O'R05]). Instead of mainly consuming information provided by (professional) service providers, ordinary users increasingly actively participate in creating online content. Users are active in social environments and the 'bloghosphere', and contribute to wikis. The use of all social media platforms, such as weblogs, photo-sharing websites, social network sites, and chat rooms, has grown significantly over the last years [Uni08].

Social media change the collection and dissemination of news, provides commercial organisations new business opportunities, and influences social life and family situation. For example, the millions of existing blogs cover nearly every topic and dissolve the boundaries between professional journalists and amateurs [Sol07]. Social network sites have an effect on the nuances in social connections, and are likely to influence the amount and quality of ties that an individual can manage [Sol07, DB04, WG99].

Personal information does not necessarily have to be shared to maintain social relations via the internet. Individuals can also act anonymously in online social environments. Many people, however, do disseminate personal information percisely because they have an interest in the creation of social capital and reputation, and because a 'display of connections' is considered important [DB04]. Because many people make use of the internet for 'social purposes', much personal information (text, video, and audio) is therefore uploaded and shared. People leave digital traces everywhere. This does not mean that these individuals upload their personal information to 'the public', in the sense that it may freely be used by others. Context still matters, even in online social media. The ease with which information can be decontextualized and used 'out of context', however, undermines the sense of 'public privacy' and can lead to reputational damage (see for instance: [Sol07]), and identity fraud. In general current web 2.0 applications are not very well tailored to help people to segregate their audiences.

3.6.2   Customer Empowerment

Another aspect of 'Web 2.0' is a change in the way customers and organisations (enterprises and governments) interact. The internet appears to intensify the relation between users and organisations. Dissatisfied consumers post their grievances on discussion fora and blogs that can be read by fellow consumers. Enterprises increasingly monitor these media and actively engage in them in order to try to manage negative scenarios regarding their reputation. Moreover, technologies make it possible to use and process the ideas and suggestions of customers directly into the process of innovation, in line with managerial trends like 'open innovation' and 'democratic innovation[1].

---

[1] Terms that were introduced by Henry Chesbrough and Eric von Hippel.

Via the internet, organisations can empower their customers, which creates an incentive to construct business models around (the knowledge of) the user. Hardware and software vendors, for instance, all have knowledge bases that are fed by their own staff as well as by users of their products. Information from users can be a key asset for organisations. The internet makes it possible to apply the 'wisdom of the crowd' to the benefit of the organisation, which means that collective intelligence can provide better insight in the requirements for services and products that need to be developed.

However, customer empowerment can also lead to more personalisation and personal data collection. These data can not only lead to better (tailored) products, but can also be used for the purposes of data mining, targeted advertisement, and discrimination.

Electronic services are provided on a global scale (web browsers need no passport to travel to different countries) and includes anything from health services (like providing medical records and medical information) to online gaming. This means that (personal) data relating to a rich set of activities flows across the globe crossing jurisdictions and policies regarding the collection and use of personal data and involving private and public entities.

3.6.3   Identity-Related Crime and Misbehaviour

The difficulty in properly identifying both individuals and organisations online has also drawn the attention of criminals. Online identities are valuable for criminals and people with harmful intent. Technologies have increased the opportunities for 'identity theft', 'identity fraud', and 'identity deception' (for definitions of the terms see for instance [KL06, KLM$^+$08]), because online identities are used in disembodied environments. The individual increasingly is physically absent when identification or authentication occurs.

Technological developments seem to have made it easier and profitable to abuse identities [MWB$^+$04]. Online financial services, for instance, have become a main target of cybercriminals (see [APW07]). Especially in the United States, identity fraud is a prevalent and fast growing form of crime [WF08, BMK07], and it has been assumed that also for Europe identity fraud is growing, even though less statistics are available for this region [LGM$^+$05]. The economic loss as a result of ID fraud for enterprises is significant [MWB$^+$04], but the negative effects do not stop there. ID-fraud can also seriously affect the trust of consumers in online services.

Identity abuse is, however, even more unpleasant for the individual. The economic loss resulting of ID abuse is often not the individual's biggest concern, but rather reputational damage, confusion, burden of proof, and the restoration of damages done are. The side effects of identity abuse may furthermore extend for years, for instance in the exclusion of services, accusations, or stigmatisation [MWB$^+$04].

One of the most popular methods of committing ID fraud is 'phishing'. Phishing concerns tricking people to reveal their confidential information by

luring them to websites that resemble those of genuine entities where the user may have an account, or sending them e-mails 'on behalf' of such entities. The collected information can then be used by criminals to make purchases, or launder and transfer money [Oll04]. Especially in the US, phishing costs companies billions, and has led to 'numerous consumer alerts and the creation of industry working groups' [EI06, P. 58].

Criminal abuse of identities is not the only form of abuse. Identities can also be abused for activities such as bullying and betrayal. On weblogs and social network sites some people may for instance intentionally reveal another user's identity or use another user's identity for the purpose of deception or manipulation [DB04]. With wrong or revealed identities, people can provoke violent reactions, destroy the integrity of an online environment, and intimidate others.[2]

The use of the internet for the purposes of criminal activities, manipulation, or deception highlights a need for thinking about accountability or identifiability of individuals in specific contexts. Moreover, the potential use of the internet for terrorist activities or activism may even further intensify the 'call for accountability' on the internet. However, such a call for accountability can also lead to superfluous surveillance and supervision, because technologies also provide instruments for constraint, control, deception, and criminality. IdM systems have a function in the creation of the appropriate levels of accountability and freedom in online contexts. The increasing use and abuse of identities furthers the need for IdM systems which have the features that facilitate such a balancing act. It is a challenge to create IdM systems that allow for accountability, without the possibility of identity abuse, and without eroding the necessary level of privacy.

3.6.4   The Expanding Internet: Always-On and Everywhere

Internet penetration and the amount of households with a computer is increasing rapidly in Europe (see [Soc07]). People also spend more time online. The use of internet already overtakes the use of television amongst young people[3], and a significant amount of users spends more than 16 hours online per week (see:[EIA07]). However, at the same time, many people seem to be concerned about the amount of personal data they leave on the internet. The amount of digital data held on every person, are exploding [Hen08], yet only a minority of internet users employ tools that increase data security [Org08].

---

[2] Famous is the Megan Meier case on MySpace. Megan Meier committed suicide after a friend, Josh Evans, a false identity allegedly created by Megan's neighbour Lori Drew, wrote that the world would be better off without Megan. See, for instance: http://archives.chicagotribune.com/2008/may/15/nation/chi-megan-meier-myspace-080515-ht.

[3] Which is emphasized by a recent IDC study, see 'IDC Finds Online Consumers Spend Almost Twice as Much Time Using the Internet as Watching TV' from 19 Feb 2008 on http://www.idc.com.

The increasing use of the internet will lead to a higher dependency on its infrastructure and on the services it facilitates. For some, the internet is a means to be 'always-on'. For mobile phones this is already the case for most users. The boundaries between work and private life diminish, many people leave their computers on and check their (work related) email in the evening and during weekends. Vice versa, private affairs are also conducted in the workplace; workers do visit websites for private purposes during working hours.

Mobile phones no longer are just phones, many are smart phones. They contain proper web browsers and email clients, and judging from the popularity of the Apple iPhone, this addition to appliances appears to be the best thing since sliced bread. Smart phones will likely increase the amount of time people spend online, which potentially means a further increase in the amount data trails people leave online. Given the fact that many smart phones also contain capabilities for determining the location of device (by GPS), which supplements the server side capabilities to locate devices (by GSM/GPRS or by WiFi positioning), the data trails can even be enriched by location data. Therefore, not only the user's behaviour, but also the location where this behaviour is exhibited can increasingly be monitored.

3.6.5   The Internet of Things and the Citizens of Tomorrow

In 2005, the ITU prepared a report on 'The Internet of Things', describing an evolution towards next generation 'always on' communications. We are moving from today's era of people-to-machines communication, from conventional Internet and mobile phones, to the era of machine-to-machine communication: the Internet of Things. In this new type of communication, new technologies, such as RFID, will enable the creation of networks with always interconnected devices. There are innumerous functions these 'things' will be able to perform. They will be able to "direct their transport, adapt to their respective environments, self-configure, self-maintain, self-repair, and eventually even play a role in their own disposal" [RFI08, p.3].

The Internet of Things will have radical effects on the way we interact with technology. Nowadays we are aware that we turn on our laptop or TV, the internet of things changes this. "It is all about making technology ubiquitous" [Sri06]. Ubiquitous computing may make individuals less aware that data is being disclosed and collected, much like many people are increasingly unaware of the camera surveillance that is becoming common in European cities.

In the today's world, the ratio of radios to humans is almost 1 to 1. The vision of the Internet of Things will challenge the very foundation of this landscape. In scenarios where even devices such as toothbrushes indicate electronically to remote devices that they need to be charged or when each light bulb in your house has a unique identifier, the ratio of radios to humans could easily exceed 1.000 to 1 [Sri06].

In the way to the networked era of the Internet of Things, also major changes are taking place with regard to identification documents. Electronic identification documents (ID documents) are seen as a necessary upgrades of important paper ones. RFID chips are chosen by the International Civil Aviation Organization and the European Union as the storage medium for data on the ID document holder. These chips have sufficient storage capacity to store biometric images and they are believed to ease the identity checks and enhance security. The equipment of ID documents with RFID chips is claimed to reduce fraud and prevent identity theft, as the ID document will not be easily tampered with. Furthermore the limiting of human inspection of the documents would help decrease the amount of errors made in the process.

The privacy and security risks that arise from the vast deployment of electronic ID documents are easily neglected. The RFID chips facilitate continuous tracking and tracing of individuals.[4] Unauthorised reading can not be ruled out and enormous databases with sensitive information about the individuals are expected to be created. The European electronic passport is already a reality and a many initiatives are currently ongoing regarding the introduction of electronic identity cards in Europe and several US States.

Besides RFID and similar technologies, the use of biometrics as identifiers is increasing dramatically. There is a transition from the traditional method of identifying yourself via something you have (key) or something you know (PIN) to something you are. A part of ones body is used as the means of identification and is the 'key' that allows her to have access to a restricted area, to operate a machine or to secure information.

3.6.6 Identifying the Individual in the Era of the Internet of Things

The Internet of Things depends on unique identifiers that will allow every-'thing' to communicate. But will every-'thing' qualify also as personal data? Will every-'thing' be linked to an individual? Will our perception of personal data need to change in order to tackle the challenges posed by this new situation?

The European legislation on data protection applies when processing of personal data is entailed. According to Article 2(a) of the Data Protection Directive personal data shall mean "any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity." Therefore in order to

---

[4] The European biometric passports do implement access control security measures, but these are not unbreakable as various studies have shown (see for instance http://www.guardian.co.uk/technology/2006/nov/17/news.homeaffairs for a story about the UK passport).

define whether some information qualifies as personal data, we need to assess firstly if the processed data relate to a natural person, and secondly whether the data relate to an individual who is identified or identifiable [PN07]. The latter question is the one that stimulates vivid discussions.

When information refers directly to an individual, such as his name, age, nationality etc., it is beyond doubt that it qualifies as personal data. The qualification is more challenging when the information can not be directly linked to a natural person, i.e. when the person is only "identifiable". Recital 26 of the data protection directive reads that in deciding whether data could be used to identify a particular person "account should be taken of all the means *likely reasonably* to be used either by the controller or by any other person to identify the said person" (emphasis added). Thus the recital sets two criteria for identifiability: the probability and the difficulty that tend to be interlinked [Byg02].

The national legislation of the European Member States and their interpretation by the national Data Protection Authorities construe the concept of identifiability in different ways. The data protection laws of France, Belgium and Sweden, for instance, have adopted a broad interpretation of the concept of personal data, rendering any information as personal data if an individual can be identified, regardless of the technical or legal difficulties in determining the identity of the individual. The German legislation, on the other hand, has adopted a more pragmatic approach to the notion of identifiability. The German Federal Data Protection Law in article 3(6) defines the notion of 'Anonymisation' as follows: "Rendering anonymous' means the modification of personal data so that the information concerning personal or material circumstances can no longer or only with a disproportionate amount of time, expense and labour be attributed to an identified or identifiable individual". The definition of anonymisation allows the deduction of the following *argumentum a contrario*: personal data are information that can be attributed to an identified or identifiable individual without a disproportionate amount of time, expense and labour.

These issues are not merely semantic battles for cocktail receptions. The 'battle' surrounding the question whether IP addresses are personal data between search engine providers (such as Google) and the European data protection authorities is centered around this issue. The Article 29 Working Party in its opinion on IPv6 sustained that IP addresses attributed to Internet users are personal data [Par02]. The same opinion was supported a few years later, where the Article 29 Working Party confirmed its opinion that IP addresses are personal data and noted that "unless the Internet Service Provider is in a position to distinguish with absolute certainty that the data correspond to users that cannot be identified, it will have to treat all IP information as personal data, to be on the safe side" [Par07].

However opposite opinions have also been expressed, presenting significant argumentation. Google, by means of its Chief Privacy Officer, Peter Fleischer has taken the position that IP addresses are not personal data (most of the

time).[5] Fleischer quotes the Secretary for Home Affairs of Hong Kong (Dr Patrick Ho), who maintains that: "An Internet Protocol (IP) address is a specific machine address assigned by the web surfer's Internet Service Provider (ISP) to a user's computer and is therefore unique to a specific computer. An IP address alone can neither reveal the exact location of the computer concerned nor the identity of the computer user. As such, the Privacy Commissioner for Personal Data (PC) considers that an IP address does not appear to be caught within the definition of "personal data" under the PDP."[6]. Although it is obvious that Hong Kong does not fall under European law, the argument expressed by Dr Ho can be valid in the current debate on IP addresses in Europe.

IP addresses will be of seminal importance in the Internet of Things era, as every little 'thing' will have an IP address that will allow it to be networked and interconnected. However it will become even more difficult for an ISP "to distinguish with absolute certainty that the [IP] data correspond to users that cannot be identified" [Par07], as required by the Article 29 Working Party. The example of IP addresses clearly illustrates the difficulties in defining whether a piece of information shall be considered as personal data or not.

3.7   Conclusion

This chapter has provided a first glance at the identity management landscape. It has introduced two different perspectives on identity management, an enterprise view and an individual view. The enterprise view concentrates on access control to resources that is usually implemented as a system of user accounts. Each account specifies which user is entitled to which services. Identity management in this perspective is closely tied to Identification, Authentication and Autorisation. The individual perspective, on the other hand, is based on the way individuals manage their identity in everyday life. Identity in this view relates to the way individuals present themselves to others and how others view them. As people engage in different (kinds of) relationships, they display different aspects of their identity. What is shown in the private setting of the family differs from what is shown in the workplace or during shopping. Identity management in this view is (unconsciously) deciding what image of self to show to others in a specific context. Individuals may present themselves as the same across contexts (I may tell my employer that I am indeed the famous tennis player by the same name) or as different (I may not tell my grocer that I work in Tilburg, even though he has seen a picture of me on the website of Tilburg University).

---

[5] See for instance his blogspot: http://peterfleischer.blogspot.com/2007/02/are-ip-addresses-personal-data.html.

[6] http://www.info.gov.hk/gia/general/200605/03/P200605030211.htm, as quoted on Peter Fleischer's blog (Chief Privacy Advisor of Google).

Context shapes how identity is constructed and maintained. The identity of an individual can be said to consist of the sum of various partial identities displayed in the different contexts. This does not, however, mean that all data related to these various partial identities can be combined into 'one' identity. Data associated to the various partial identities is contextual and therefore the combination of data may lead to seemingly inconsistent pictures.

Identity management developments are until recently, driven by the enterprise perspective. Originally each entity requiring access control developed and maintained their own solution for implementing access control. The result of this has been a plethora of fragmented and incompatible IdM solutions. For individuals the consequence of this landscape is that they have many online identities that are composed of similar data that was disclosed to each and every of the enterprises. Furthermore, the user has little control over the identity they want to present to the various enterprises. Their freedom to present themselves as the same or different is limited by the restrictions imposed on them by the IdM systems.

In recent years, a move towards identity federation can be observed. Enterprises collaborate and design systems that allow interoperable identity provisioning and access control. These developments primarily solve enterprise needs because these systems lower their expenses in setting up and maintaing IdM systems. Also the user benefits from the single sign on functionality offered by federated IdM, but the lack of control over the identities to be used largely remains.

A step further is the move towards user-centric IdM where the individual is at the steering wheel. The individual creates and maintains her online identities and populates these with credentials obtained from the various identity providers. The level of control over the presentation of self can be significant in these systems.

Not only the unification of the enterprise perspective on IdM with an individual perspective is challenging. We have also described a number of technological developments that complicate identity management. Users are changing from consumers to producers of content (Web 2.0). They actively engage in social networks, blogs and wiki's and disclose data on the go. Furthermore, technology is increasingly becoming pervasive and ubiquitous. More and more device are networked and connected. This raises questions regarding the identification of things in what is called the Internet of Things. As things are used by humans, there clearly is a link to the identification of humans and to identity management of humans and things. The developments make clear that the existing concepts on which data protection and privacy regulation is built no longer self evidently adequate.

The IdM landscape is evolving rapidly. Until recently privacy concerns hardly have played a role here. As we will argue in the following chapters, this needs to change and we will show that this is indeed possible.

4

The Need for Privacy-Enhancing Identity Management

Bart Priem[1], Ronald Leenes[1], Alea Fairchild[1], and Eleni Kosta[2]

[1] Tilburg University
[2] KU Leuven

4.1   Introduction

The previous chapter described current developments in identity management. Identity management systems are moving away from enterprise centric 'silo' systems towards federated and user-centric systems. The traditional single enterprise solutions with their identity data 'silo's' are becoming obsolete because of the collaboration between service providers and because they are burdensome for both the individual and for organisations. Current developments towards single sign on and identity federation do acknowledge the complaints about the inconvenience of traditional identity management systems. They do, however, still mainly focus on the enterprise identity management needs: access control to resources. On the forefront of IdM developments we observe projects where the individual is increasingly placed center stage rather than enterprises.

User-centric identity management should take the social and individual perspective on identity and identity management into account. Identity management in this view should see to the diversity and autonomy of individuals. Individuals should be able to decide how to present themselves in different contexts. This means being able to use the same identity in different contexts and using different identities in the same context. Proper user-centric identity management therefore takes privacy into account. Users, within certain bounds, need to be able to keep different audiences separated and determine what they reveal of themselves. This chapter will make a case for taking privacy seriously in identity management by describing the necessity

of privacy from three perspectives: the individual, organisational, and societal perspective.

4.2 Individual Perspective

Individuals engage in different social and economic relations online. What they reveal of themselves is partly determined by the image of themselves they want to convey, and partly determined by others. What information is provided in the various different relations is diverse. In practice, the result of all these interactions is that each individual explicitly of implicitly creates many online partial identities or digital personae ([Cla94]) over time.

Many people may be unaware that the creation and maintenance of these digital persona is identity management, even though in practice the 'management' part is fairly restricted. Individuals usually create an identity when they register for a user account at some service provider. This usually requires them to complete an online form where the service provider determines what constitutes the online identity. This is usually the endpoint of online identity management for the user. Online identity management from the perspective of the enterprise entails much more because the enterprise as part of the enrolment procedure collects considerable amounts of personal information for reasons to be discussed later (see Section 4.3). The user does have to manage their online identities though in the sense that they have to keep track of all the usernames and passwords associated to their different online identities (accounts).

The kind and amount of personal data the user is required to provide on registration is similar for many new online service they engage in. By and large, people have to provide the same contact information for every online service. This is burdensome for the user. Many users faithfully provide the data requested, but there are also significant numbers of internet users that provide false data that barely meets the requirements on the forms.[1] For instance, when a phone number is required, users enter a number that passes the site's test for valid phone numbers, or for (confirmation) email addresses garbage can sites, such as spam.la are used. This data pollution by incorrect data means that also service providers should not be satisfied by this kind of identity management.

Given the plethora of online services used by the average online user, it is no wonder that users resort to tools that make the management of their online identities easier. Many browsers can remember usernames and passwords and can assist in completing online forms (form fillers). Also more advanced tools are on the market. However, an emphasis on (in)convenience obfuscates the fact that identity plays a crucial role for the individual in the

---

[1] See for instance the PRIME survey ([OL08]) that shows that about 45% of the respondents sometimes provide false data when they don't consider the data relevant in the given context.

information society and that the forms and scale of identity-related crimes that harm the individual are changing significantly (see [KL06]). The collection and storage of identity information in databases provides, for instance, an immanent risk of ID fraud, customer profiling, data manipulation, data-mining, target advertisement, data loss, and discrimination (see for instance [Gan93, Lyo01, DG02, Lyo04, Les99]). Discovering identity fraud, even though the abuse of identity information can lead to identity deception, discrimination, financial damage, identity confusion, and reputational damage (see [Don98, Sol07]) is sometimes difficult and more frequently it is difficult to get hold of the entity that can resolve the issue. Proof of fraud is usually also difficult (see, e.g., [LGM$^+$05]) while the burden of proof often also rests on the wrong shoulders. The affected individual usually has to prove that she is the victim of identity fraud, while such proof is much easier to establish by someone else, such as a bank in case of fraudulent transactions. As a result, damages are usually difficult to undo by the individual.

Convenience is therefore only one of the reasons to invest in identity management. The risks of fraudulent use of identity data are equally important. Embedding privacy into identity management systems is necessary to protect the individual and their digital personae.

In the following sections we discuss four aspects of the online world that further underpin a need for privacy aware or privacy-enhanced identity management. First, privacy in IdM systems decreases possible abuse imminent in the *power imbalance* between the individual (the data subject) and the user of the identity information (the data controller). Second, privacy enhances the options to develop and maintain meaningful *relations* on the internet; something which is difficult at present due to the ease with which information can be copied, transferred, and used 'out of context' [Sol07]. Third, being able to conceal specific identity information from the gaze of others, promotes *personal development*. Finally, integrating privacy in IdM systems is thought to have a positive effect on the *behaviour, health, and emotions* of the individual.

4.2.1   Power Imbalance

Identity management systems facilitate one or more parties to have identity information concerning an individual at their disposal. These data may be essential for establishing and maintaining trust in the relation and for providing services to the individual: name and address are usually necessary to deliver tangible goods, the telephone number and e-mail address may be used to contact a consumer in the case a delivery is delayed, and credit card data may be necessary for payment purposes.

However, the collection and use of these personal data also make data subjects vulnerable to current and future actions of others. This vulnerability arises both from the *collection* of personal data and from the actual *use* of these data. The collection itself presents issues because a lack of transparency with regard to the collection of the data may already have a disciplining and

normalising effect on the users of online services [Fou77]. People behave as they expect they should when they know they are observed. After collection issues arise because the data subject who provides the data cannot predict whether a data controller will commit fraud, lose data, sell data to others, or make wrongful judgements on the basis of the disclosed information.

User control and privacy protecting measures may help limit the vulnerability of data subjects caused by the power asymmetry, because it can both restrict the accumulation and the use of personal information.

When privacy concerns are acknowledged by data controllers, individuals will have the possibility to shield information from contexts in which this information could potentially be abused. Because power abuse may originate from virtually any actor in society, it is important that IdM systems not only facilitate keeping certain personal data private (shielded from the public at large), but also provide ways to control which data is provided in specific contexts and relations. Leveling the power imbalance by means of privacy enhancing tools also means that individuals should be able to take actions against abuse of private information. Privacy protection therefore means empowering individuals before and after data disclosure.

Addressing power imbalances by respecting privacy concerns serves important individual values, such as *human dignity*, *autonomy* and *freedom*. Human dignity relates to respecting the individual and giving them the possibility to partially control their image to others [Whi04]. Respecting dignity in an informational contexts means that inappropriate use of information by others should be prevented and that situations that could lead to embarrassment, unwanted exposure, and humiliation are restrained [KL05]. Autonomy means that people can make their own choices with regard to the disclosure of identity information in different contexts and relations. Freedom relates to the fact that people should be free to make choices regarding their presentation of self and that their personal sphere, or intimate context, is respected in relationships; their identity creation and development must not be intruded by third parties (e.g., by means of wiretapping, eavesdropping, or cracking). Autonomy and freedom of individuals are under pressure in online contexts because the disclosure of personal information may be observed and behaviour of citizens, consumers, and relatives, may be monitored which potentially limits their options to make their own choices in life. Examples here are the practices of social sorting and data mining, which are techniques that rely on the collection and analysis of personal data and that make it possible to judge, assess, and exclude groups and individuals [Gan93, Lyo01, DG02, Lyo04]. Judgements may take place on the basis of incomplete or incorrect information, or information that was disseminated for other purposes (decontextualised) and judged 'out of context' [Sol07, Gan93]. Only when power between a data subject and a data controller is balanced (by means of privacy protection), wrongful and 'out of context' judgements can be addressed by the individual.

Of course, the empowerment of individuals must not result in absolute control of a person over his personal data because this would imply a

complete dependence of data controllers on data subjects which negatively affects the free flow of information. Empowering the data subject by giving them privacy protection merely implies correcting imbalances in the power between data subject and data controller in the construction and development of their identities.

4.2.2   Relations

Being able to maintain different identities plays an important role in the development and maintenance of human relationships, because different kinds of relationships impose different rules regarding the participants which has a bearing on the information that is seen as appropriate to be disclosed. One person can, for example, be a customer, father, salesperson, voter, and amateur football player. All these situations require different behaviour of the individual. Generally it would be deemed fairly inappropriate to reveal intimate details of ones love life to a teller in a supermarket.

The construction and maintenance of roles can be characterised as a 'theatrical performance' [Gof59], in which one plays different roles to different audiences. Roles and audiences need to be segregated because otherwise the possibilities to maintain different kinds of relationships will vanish ([Rac75]). Intimate relationships are impossible if everything that is said and done within such a relationship would be public knowledge. A certain amount of privacy, or control over what is presented to others, is necessary for offline and online relations [Int97]. Privacy creates the preconditions for love, friendship, accountability, and trust, without which relationships would be inconceivable [Fri68, Int97]. Performances also need to be insulated from the activities that occur 'backstage' out of sight of specific audience to provide the individual a possibility to adapt his or her role to changing circumstances. Privacy is functional to this 'insulation' and 'segregation'.

Identity management systems play a role in the establishment and maintenance of meaningful and intimate relations and therefore need to incorporate privacy features by providing the possibilities to segregate and insulate partial identities in relations.

Privacy in this respect is not an absolute value, but provides a level of identity-building in relations that is 'free from unreasonable constraints' ([Agr97, Hil06, p.7]). This means that the control over how individuals present (and represent) themselves in their relations should not needlessly be affected by others. In relation to online identity management, this means that people in the online environment should be able to create and maintain characters (identities) for their 'roles' as a customer, father, salesperson, voter, and amateur football player, without conflating the data associated to these different roles, comparable to how this works in the offline environment. To facilitate this, the construction and maintenance of digital personae should be under control of the individual, and not be limited by unnecessary constraints imposed by the identity management system or the identity providers hosting this system.

Furthermore, the IdM system needs to provide individuals the private realm in which to construct and assess identities. Lack of such a privacy-feature ultimately leads to one-dimensional online relationships [Gav80, DB04].

Maintaining human relations also requires people to be able to temporarily withdraw from such relations. Otherwise, these relations would be unbearable, and could lead to antisocial behaviour, confusion, irritation, or even hostility [Sch68]. Although the online environment differs from the offline environment, people also need to be able to withdraw from relations online. Online, this will mainly relate to the storage of identities and the access of others to these identities. Furthermore, withdrawal may also be required to provide people the opportunity of a 'fresh start' and a level of 'forgetfulness' which is an important feature in the real world, but which is not provided in the online world by default [BJ02]. Identity deletion is therefore an function that should have a place in online identity management systems.

4.2.3   Personal Development

Privacy features are also essential to provide individuals the means to autonomously, that is without unreasonable interference by others, construct their (online) identities and deploy them in different relations. Privacy aware or enhancing IdM systems should provide an online equivalent of the backstage environment of theatrical performances, as described by Goffman in 'The Presentation of The Self in Everyday Life' [Gof59]. An 'online backstage environment' can for instance be provided by allowing people to act anonymously or pseudonymously online, so that an individual can develop himself and his identity without the risks of being exposed whilst learning and making mistakes.[2] Visibility of mistakes can result in significant reputational damage or torment, if these mistakes can be linked to a specific individual. Moreover, if all mistakes of a person would be potentially visible, technology facilitated creativity, experimentation, and learning would severely be undermined.

Circumstances and contexts change over time. Individuals evolve over time even though they stay the same. The may feel the need to change their digital persona (their old characterisations) over time accordingly. People, lives, attitudes, and opinions change and therefore the individual's online representations should also reflect these changes [WP205]. The adaption and updating of partial identities requires an environment in which these partial identities can be assessed, defined, examined, and aligned with current circumstances. An IdM system should provide these options..

---

[2] The online virtual world Second Life, for instance offers its Residents the option to instantiate alternate accounts (Alts) that allow the user to switch from a clear identity to an anonymous identity unlinkable (for other users) to their primary Second Life identity.

4.2.4   Behaviour, Health, and Emotions

Personal information can easily be shared, copied, and transferred, which makes it potentially possible to expose information to the wrong and/or to too many entities. The audiences of internet services such as weblogs, e-mail,[3] and social network sites may start small, but can potentially be global [Sol07]. People and organisations can easily interact with large numbers of people, also when it concerns the use of damaging or wrong information [Sol07]. Furthermore, time and space lose their significance with respect to confining information [Lyo01]. Harmful content relating to an individual can therefore be exposed to anybody, at any time, by everyone. Because of this, online gossip and online bullying can potentially have more serious effects in the online world than they do offline.[4]

The changing influence of time, space, and disembodiment in online life also affects the possibilities for surveillance and scrutiny of others. With regards to surveillance and exposure, the internet is a true global village, facilitating continuous interaction of our digital personae [Sol07, Lyo01]. Even though this exposure does not relate to our own bodies or territory, we still are potentially exposed to large groups of people and unknown organisations.

It is difficult to determine the behavioural, emotional, and health effects of this online exposure, but it can be noted that a lack of privacy in systems of identity management easily leads to overexposure or unwanted exposure of a human being and their digital personae.

Excessive contact with others may lead to irritation, stress, or disappointment, especially when control on attendance in an online environment is lost (see, e.g., [Alt75, Sch68, Wes67]). Just as we react to dense and crowded offline situations, e.g., by tuning one's voice down, hiding feelings, and experiencing anxiety or stress (see, e.g., [Alt75]), the online environment is likely to influence human behaviour, as currently an individual has little control over his or her exposure and has not much foresight with regard to the context and attendants in online interactions.

Privacy is furthermore considered necessary for the individual to have some kind of individual 'safety valve' [Wes67, p.35]. Everyday life creates tensions and stress, which occasionally need to be vented. Westin claims that this need for emotional release and relaxation is important both for the physical and psychological health of the individual [Wes67, p.34]. Moments of relaxation or emotional release can lie in being anonymous, or in playing a specific role (e.g., a character in an online game, or being a pseudonymous blogger). However,

---

[3] See James Grimmelmann's excellent account of how an email account of the World Economic Forum by Laurie Garrett to her friends spirals out of control [Gri08].

[4] An example of cyberbullying is the Megan Meier case. Megan Meier committed suicide after a friend, Josh Evans, a false identity allegedly created by Megan's neighbour Lori Drew, wrote that the world would be better off without Megan. See, for instance: http://archives.chicagotribune.com/2008/may/15/nation/chi-megan-meier-myspace-080515-ht.

one would only be able to have these moments if a certain level of privacy is assured. Thus, IdM systems ought to provide a possibility for emotional release without this behaviour being exposed to others. If IdM systems lack this feature, the possibilities to cope with shock, sorrow, or irritation in life would become difficult in the online environment [Mar03, Wes67].

Privacy of the individual in the online environment also limits the pressure on the individual to exhibit 'normalized' behaviour in the light of surveillance. Lack of privacy may reduce intimate and spontaneous interactions with, for instance, close friends because there is a constant possibility of 'third parties' being present (see for instance [Fou77] on the effects of panoptic surveillance). An omnipresence of third parties could actually mean that intimate relations would not be possible at all, because these would make the relations insignificant [Rac75]. Relations that are maintained on the internet may be heading towards this situation, because privacy cannot easily be obtained which can be illustrated by the fact that for instance in online social networks 'no distinction can be made between a close relative and a near stranger' [DB04, p.72].

The (possible) presence of a 'third party' in relations is also an instrument that can be used to *discipline* the individual on the internet in the broadest sense. This may occur with regard to consumer transactions, the working environment, and citizen behaviour, alike. Norms are not only set by governments, also commercial entities require their customers to answer to certain criteria, just like employers, family and friends. In fact, the internet without privacy protection provides many actors an architecture to control and discipline the individual. In other words: the online environment facilitates a 'virtual panopticon' in which potential surveillance may force individuals to adjust their behaviour towards the norms set by others [Fou77, KL05, Gan93, Int97]. When IdM systems lack privacy, we may therefore question what in the end will remain of the 'true individual' [Int97, p.273].

4.3 Organisational Perspective

In the realm of technology facilitated service delivery, identity plays an important role for business. Identification, authentication, and authorisation play a central role to access control to services: controlling that only those entitled to a certain service are able to obtain it. IdM systems are significant to both electronic business- and government solutions: they are 'key business enablers' [CMBG$^+$02]. But it is also in the interest of businesses and governments to implement privacy in their IdM systems.

4.3.1 Business

Traditional drivers for implementing privacy features into enterprise systems from an information economic perspective include the following:

- compliance with legal obligation,
- fear of reputational damage from privacy failure,
- the need to generate trust with clientele, and
- promotion of a good corporate practice.

Yet if these drivers were truly present, then privacy enhancing technologies would be much more widespread than they are.

A reason for this discrepancy is that reality turns out to be more complex. In fact the traditional drivers are considered by the experts as insufficient, leading to strong doubts regarding the presence of a 'well-structured business case for privacy' (see also chapter 7).

- Compliance is not taken seriously enough to be retained as a driver, since there are so few investigations dealing with Privacy practice and even fewer penalties associated with non-compliance. This may change, however, after the imposition of such heavy penalties as the ruling against Nationwide Bank in the United Kingdom who were fined nearly £1 million by the financial regulator for inadequate response to a data breach. Also in the US, as a result of the Security Breach Notification Laws implemented in 44 states already seems to have had profound effects on practices within companies. Breach notification laws drive information exchange between organizations, and within organizations themselves [Sam07].
- Reputational and Brand Damage is not seen as necessarily linked with public disclosures of privacy failures. Research and experience in this regard are not conclusive whether organizations actually experience damages to their reputations from data breaches. Many experts doubt that reported data breaches may induce a loss of customers and thus hurt companies. TJX Companies, the mammoth US retailer whose substandard security led to one of the the world's biggest credit card heists, did not seem to suffer much from the affair.[5] Some research, however has shown that negative stock market fluctuations do take place after the announcement of a breach.
- The notion of 'generating and maintaining consumer trust' as a privacy driver is a large and perhaps unwieldy goal that is never quite verifiable. While this terminology permeated much of the discussion around e-commerce in the 1990s, there is much less discourse about trust today. Privacy has not yet emerged as a 'differentiator in the marketplace' - if it were so important then certainly some organizations would make much more advertising use of their 'privacy-friendly practices'.
- There is much faith in the idea that protecting privacy is merely another way of showing good corporate practice, but it is only recently that discussions have emerged about including privacy within corporate social responsibility regimes.

---

[5] See for instance http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1278757,00.html.

Organizations do not currently understand the nature of the risks associated with by the processing of personal data. Just as it took organizations quite some time to learn about information security, some believe that this ignorance of the potential risks explains the lack of awareness and understanding about privacy and the tendency for businesses to collect and retain as much information as possible, the more so as storage costs spiral downwards. It may thus take time until this trend is re-considered and stopped, which may result from data breaches and other security concerns.

Privacy may also be seen to follow the same course as 'Total Quality Management' in the sense that taking privacy seriously may be a way of 'tightening up the ship' by providing better information management. This approach highlights that privacy may not be the 'good' that is being delivered (or sold) but instead the rise in consumer and organizational confidence is the ultimate goal.

Privacy also falls into the area between 'social responsibility' (good citizenship) and 'compliance'. When oil companies gain credit in the opinion for spending money on research into alternative fuels, this is more perceived as 'good citizenship' than as the result of a regulatory-burden (at least not yet). Privacy currently is more seen as a compliance issue and insufficiently as a good social practice. Some of the consulted experts in [WP008], however, felt that there was much room for growth in this domain, and that privacy management may eventually be seen as part of an organization's general attitude and a revealing indicator to judge it. If a firm would show negligence in the processing of personal information, this would raise questions among consumers and business partners whether this may indicate a negligent attitude ('poor citizenship') possibly spreading to other business domains of the same firm, such as staffing policies, or even the honouring of warranties.

To emphasize that good conduct in privacy matters is an important part of 'good citizenship' and as such a social goal, which would certainly contribute towards a widespread adoption of privacy practices, it could be stressed that privacy invasion is socially harmful, as it is the cause of three types of 'harms':

- the harm that is created for the individual and the consumer;
- the harm to the corporation due to the time and expenses in rectifying the root problem and its effects;
- and the harm to society as a whole due to the reduced confidence in the sector and perhaps across sectors.

Once privacy failures are emphasized on all these levels then a positive demand for privacy within organizations may emerge and become stronger with each 'privacy disaster', leading to the ultimate goal of seeing privacy as a differentiator in the marketplace.

4.3.2   Government Services

The potential of ICT's has been recognised widely by governments [Pri07a]. ICT's provide the government with the possibilities to change their internal organisation, save costs, and become more effective and efficient. Furthermore, technologies such as the internet offer significant opportunities to improve public service delivery and become more 'customer-oriented' and thus also improve the relation with the citizen. This customer-aimed perspective and the restructuring of public services by means of the technology fits well in managerial trends, like 'New Public Management' (NPM)[DL01] and 'reinventing government' [Sil01].

e-Government promises electronic public services ranging from simple information services to interaction and transaction services pertaining to online tax returns, social insurance services, and granting licenses and subsidies. The online environment can also be used for democratic participation, to empower citizens to vote, or to provide a realm in which political issues can be discussed.

However, in order to provide electronic services, government needs to have identity management in place. Proper handling of online identities and personal data here is even more important than in the private sector. Privacy issues are therefore significant factors to take into account. Citizens making use of electronic public services are not 'real customers'. Usually they have no choice to go elsewhere if the conditions or privacy policies are unfavorable; the government is a monopolist for many public services. Citizens are therefore less flexible and autonomous in their interactions with government than in commercial relations. Also because the data to be provided for certain services is mandated by law and predetermined and pre structured to a high degree. Thus, personal identity in citizen-government relations is more constrained than in other relations because citizens are obliged to make use of prescribed identity attributes, because government services are accountable for reliable, effective, and qualitative public services.

Citizens interact with government in different roles with different requirements. Citizens may act as electronic voters and should then not be linkable to their interaction in other roles, such as tax payers or traffic offenders. It is therefore important to keep these prescribed identities concealed from other contexts. On top of this, recent history has shown that implementing reliable, secure, and efficient electronic public services is difficult, and sometimes even facilitates data loss or ID theft.[6]

Despite the fact that using technology to improve government services has a long tradition, it turns out to be extremely difficult to implement ICTs in the government, partly because of the scale of the endeavors and the complexity of the services and underlying processes [Pet02]. This difficulty in managing large public IT projects threatens to undermine efforts to implement e-Government

---

[6] Like the loss of child benefit records in the U.K. in November 2007 or the exposure of personal data of millions of Chileans in May 2008.

[LT02], and the confidence of citizens in electronic public services may decrease with every new failure in such projects.

Especially identity and privacy are key challenges for e-Government [Pri07b]. If these concepts are not implemented properly, the government risks reputation damage, and loss of public trust. This is strengthened by the fact that electronic public services operate in a 'trust tension' [DGZP05]. On the one side it is necessary for the government to collect data relating to citizens to provide services, but on the other side this can increase the fears of surveillance, undesired secondary use of personal information ('function creep'), and unwanted combination of public databases like the use of tax information for social insurances.

Another complicating aspect is that the 'trust tension' in government services is magnified by the need for transparent government (freedom of information) which has tension with the need to protect personal information [Raa04].

All these aspects emphasize that IdM systems in government services need to put in effort to find the balance between the use of information that relates to the citizen's identities and the necessary level of privacy and security. Finding this balance decreases the social costs that are related to data loss, contributes to trust and reputation in the government, and increases the adoption of electronic public services. Not only should convenience be a part of customer service, but also privacy and identity.

4.4   Societal Perspective

The third perspective on the need for incorporating privacy into identity management is the societal perspective. Informational privacy is well studied from the perspective of the individual and also the organisational perspective discussed in the previous section is relatively well understood. Identity management mainly seems to occupy the space in which individuals and organisations interact. It concerns the interaction between data subject and data controller, and their respective requirements and needs and the framework provided by regulation such as Directive 95/46/EC. The latter is part of the social dimension of privacy. The Directive embodies the way Europe values the protection of personal data. Society as a whole sees informational privacy as a value to endorse and has formalized this by means of provisions dealing with the conditions under which data may be processed. There is, however, much more to the social value of privacy.

Describing the need for privacy in IdM systems based only on an individual and organisational perspective has the pitfall of trading off the importance of identity and privacy for the individual against economical or other, more 'social' values. Privacy is often regarded as an individual value, rooted in liberal thinking, and placing the individual at the centre of concern [Reg95]. The discussion regarding 'meaningful relations', 'dignity', 'autonomy', 'freedom', 'emotional release', and 'self-development' in section 4.2 are typical for this

perspective. In the public debate, however, the emphasis on this individual notion of privacy has led to a constrained debate. In this debate privacy is often placed opposed to other (competing) interests that are defined as social values, like 'security', 'economic growth', 'fraud detection', and 'law enforcement' (see for arguments from a communitarian perspective on the role of privacy, e.g, [Etz99]).

Privacy has more to it than just an individual importance. A need for privacy is not an anti-social claim of the individual to conceal unwanted behaviour, but also a set of 'social norms about how intrusive we should be into each others lives' [Sol07, p.72]. In addition, privacy is not a value that is superfluous if one has 'nothing to hide', but there is a common interest in having a certain level of privacy in society. It therefore does not conflict with social values but, in fact, *is functional to society*. Being an individual with privacy thus does not mean withdrawal and concealment from society, but being a part of it. It means engagement and participation with others in a confined context, inside a constant process of boundary control.

We will elaborate the social perspective on the need for privacy in IdM in two sections. First we will dwell on the fact that the use of personal data is to some extent influenced by social norms. Second, we elaborate on the fact that implementing privacy in IdM systems benefits both a common, a collective, and a public value in society (following the work of Priscilla Regan [Reg95]).

4.4.1   The Determination of Privacy in Social Context

A very common definition of informational privacy relates to the possibility for individuals to control the dissemination of their personal information to others (see, e.g., [Wes67, Rac75, Fri68]). This control allows people to obtain and maintain their reputation, dignity, intimacy, and autonomy. Individual control and self determination are key requirements for privacy protection. However, this does not mean that individual informational control only contributes to individual values and that society does not affect individual control over personal data. Absolute individual control is difficult to achieve by individuals, as it is difficult for them to make rational decisions with regard to privacy [AG05]. This may explain one of the contemporary privacy paradoxes: the disparity between privacy attitudes and privacy behaviour in the online world. Incomplete information, bounded rationality, difficulty to weigh privacy costs against benefits, and incomprehensibility of privacy threats seem to indicate that privacy in society cannot always be assured by the sum of all individual privacy decisions (see for instance [Sho03, AG05, Sta02, BGS05]). Social values and instruments developed by society thus need to complement the capacities of the individual. Privacy is a common interest (resembling a public good) and because the overemphasis on 'individuality' occasionally seems to turn out into a pyrrhic victory for the individual (see for instance: [Sch92, p.24]).

The dissemination of personal data is and has been governed by social norms and our personal perspective on obtaining privacy is not culturally

neutral. Individual identity-related decisions also effect other people with consequences often in the future. Individual decisions therefore effect the privacy of others. For instance, when I place a picture showing some drunken friends on my Friendster profile, this may possibly affect their chances of getting a job.

Social norms and social control (next to legal norms) are therefore necessary to limit adverse decisions of individuals. Some forms of disclosure are not done, not even on seemingly norm free environments such as social network sites. Some social circles require a different treatment of identity information than others [Sol07]. Human nature and personal identity derive themselves from different social contexts. Conformity to such contexts and conformity to the people in these contexts is necessary to be able to live together [Sch92, Gof59]. True atomistic individuals do not exist ([Reg95], citing Waltzer) and are also undesirable from a social point of view. This will also count for the use and dissemination of personal information in contexts. Both for the sake of our own identity and the identity of others in a particular context, social norms play a role in determining what is private information and what not. These social norms are important, given our imperfections as deliberators and actors and the fact that we are mutually vulnerable and occasionally unable to promote the values that matter [Sch92]. Especially in the virtual online environment which is new and evolving, this seems to be the case. Unfortunately, social norms are underdeveloped in many online contexts.

4.4.2   The Contribution of Privacy-Enhanced IdM to Society

This social value of privacy can be decomposed into a collective value, a public value, and a common value [Reg95].

Individual solutions to privacy concerns that are based on the market for personal data are often ineffective in ensuring privacy. Privacy therefore comprises a certain *collective* value. In general, individuals will have difficulties in determining what kind of information is appropriate and necessary to disclose in a specific case, for instance in obtaining a service. Often there is insufficient information or knowledge available to judge whether a service requires the personal data requested. This information asymmetry means that individuals are in bad position to trade and bargain. Markets generally only function adequately in situations of information symmetry (see for instance [Ake70]). Moreover, the market is an inefficient mechanism to assure privacy, because the economic benefits of collecting personal data are clear to commercial organisations and individuals, but the costs of losing personal data are unclear [Sta02]. For example, in many occasions the share of personal data will provide the individual with a direct access to a service. However, the negative effects of his or her actions with regard to this personal data may occur many years later. This may also partially explain why individuals do not seem to invest in individual measures to protect privacy, even though many are concerned about their privacy [Sho03].

The need for a collective approach to privacy thus stems from the fact that there is an asymmetric relation between costs and benefits as well as between the incentives of data subjects and data controllers. The issues need to be approached on a grander scale than the individual.

A further issue is that information is an extraordinary economic good. Many commercial services depend on personal information, and this information is often costly to acquire. However, after it has been collected, it is quite easily transformed, copied, and transferred to others. This increases the incentives for commercial organisations to sell and distribute their data. The extraordinary nature of information as an economic good also relates to the question of ownership and economic loss. Personal data can be shared with others, without the sharing party losing any of the intrinsic value of the information.[7] Furthermore, organisations that have stored personal information that was lawfully collected, will claim ownership of these data. However, when personal data is copied, lost, and/or 'stolen', the actual economic loss will just as well affect the data subject. In fact, the implications for data subjects will be even worse because the identity information can easily be used for purposes of identity fraud or identity abuse, which is difficult to remedy. This also explains the commotion when large amounts of personal data are lost or stolen.

Another aspect that support the view that privacy deserves to be treated as a collective good is that identity information is difficult to confine and define, which makes detailed regulation and propertization difficult. Privacy protection by means of intellectual property, for instance, has been promoted (see [Pri06b] for an overview of literature in this field), but the different notions of privacy through contexts will make it difficult to define on a regulatory level what information should be included or excluded in such a property right.

All in all, clear-cut solutions to privacy issues that are based on an economic use of personal information are difficult to develop, and '[e]conomic interests and financial damages are difficult arguments to employ when it comes to discussing the rationale and actual amount of privacy protection' [Pri06b, p.226].

A one-sided economic approach towards privacy and a propertization of privacy is also on uneven footing with personality and human dignity. Dignity and reputation are core themes when it comes to privacy protection, especially in Europe (see [Whi04]). This also explains the European human rights approach towards privacy.[8] Privacy should protect *people*, not (in)tangibles.[9]

Data sharing by individuals is often involuntary [Reg95]. Individual decisions regarding privacy are not always a statement of free will [Pri06b]; data collectors often empliy a 'take-it-or-leave-it' approach, while users have little

---

[7] This also makes most forms of identity theft peculiar, nothing is stolen, the data is merely copied [LGM$^+$05].

[8] Art. 8 ECHR.

[9] Katz v. United States, 389 U.S. 347 (1967).

choice to go elsewhere [Sta02]. For the efficiency of many services in society, like healthcare, social insurances, and mortgages, personal data of *all the individuals* that make use of these services is required. In addition, the dissemination of personal data does not always relate to one single individual, but could also effect the lives of close relatives, family members, and other members of the social circle one is operating in.[10] Hence, having privacy or giving away personal information can affect the lives of others.

Another collective aspect of privacy lies in its enforcement. Privacy infringements can be committed without knowledge of the data subject and are difficult to undo and repair. The actions an individual has to take to enforce their privacy sometimes call for considerable efforts of the individual, often with a paradoxical outcome for this person: more exposure. Therefore, a collective approach towards privacy enforcement is often a more suitable approach than an individual approach.

In summary, the market will not produce privacy by itself, and individuals cannot protect their privacy by their own devices. This requires a collective approach to privacy. Privacy is a collective value comparable to clean air: we all benefit from its existence and when it is constrained not only individuals but society as a whole will be harmed. Because of this, privacy needs to be obtained in IdM systems because it affects *everyone*, not just the 'atomistic' individual.

The second social aspect of privacy is that it has a *common* value, in the sense that privacy is a shared interest, even though it needs to be defined individually. It seems that 'concern about privacy is evidenced in all societies' [Gav80, p.445]. Moreover, the origin of privacy has even been related to the aspects of 'social distance' and 'personal distance' that are present in the animal world, which demonstrates that privacy has some kind of intrinsic, common value [Wes67, Alt75].

Another common aspect of privacy is that privacy is one of the building blocks of society. Hence, the choice about what kind of society we want, determines the general level of privacy that is required. If we want a society in which people can have a meaningful life, diverse relations, and in which they can develop themselves freely, privacy ought to be provided to every single individual [Gav80]. Privacy, as said, also facilitates individuals to be different from each other. It promotes social pluralism and tolerance, because all people would have the equal possibility and opportunity to have a private realm. This makes privacy a condition for equality, and 'enables the development of the type of individual that forms the basis of a certain type of society' [Reg95, p.222]. For example, in many societies aspects of trust, accountability, friendship, and cohesion are important values. Privacy provides the context for these

---

[10] Consider information obtained from DNA material which reveals information about genetic diseases which may be present in other family members without them knowing.

values, and if a society would renounce the value of privacy ultimately those other values would erode as well [Int97, Fri68].

The internet is evolving to become an environment in which every actor in society is required to participate. In the future, it may become an essential part of our everyday life, a common medium. Because of this, identity management will touch upon almost every individual in the future, making benefits but also risks of online identity management ubiquitous. Surveys amongst internet users show that almost all citizens are concerned about their privacy in the online environment, even though there may be different levels of concern.[11] Hence, we can assume that not many people approve the idea of a society or an online environment in which there is no privacy and in which complete surveillance is the standard.

Another contribution of privacy lies in its *public value* to society. To a large extent, this relates to the organisation of the democratic political system. A political system that uses public roles, attaches importance to free speech, and has an honest electoral system which provides its citizens privacy in certain contexts. Hence, if we want technology and identity management to contribute to the democratic political system and a democratic society, privacy needs to be integrated, also into identity management systems. *Without* this, IdM systems may even become detrimental to democracy and the public realm. The online environment provides considerable opportunities to enhance democracy, but it is important to note that these opportunities can only be exploited when a level of privacy is guaranteed for the citizen.

Important institutions in a democracy are freedom of speech and freedom of thought. Citizens need to be able to assess the acts of their representatives, and be able to address their views on public policy. This requires a private sphere. State intrusion in the assessment of public policy is undesirable because it may restrain the citizen, just as much as infringing on the exchange of opinions. For democracy, it is necessary that people can vent their opinions protected, without consequences, confined from other contexts, and — in some occasions — anonymously. Privacy ensures these guarantees. Hence, even though privacy sometimes opposes free speech, in the public realm privacy is a condition for free speech, which can be compared with the level of privacy that is built-in in a system for anonymous voting.

'[T]he government should be sensitive to unreasonable constraints on identity building' [Hil06, p.56]. This means that, for example, the targeting of political messages to specific public individuals and the practices of social sorting should not be within reach of the government. However, the online environment can provide much politically interesting data which can be used

---

[11] A 2005 Eurobarometer report showed that 94% of EU citizens believed that protecting information about private life from misuse and exploitation would be important for society in ten years time [Eur05, p.64]. See also the PRIME survey results [OL08].

to normalise individuals and influence their civil identities. This underlines the importance of privacy in IdM systems in democratic contexts.

The actual maintenance and development of the public and public roles in society depends on the level of privacy that is provided to the individual. For democracy, it is necessary that individuals engage in the public debate, and that these people employ public characters. The probability of people engaging in public debate and public roles is higher when such roles are used in the right context and when aspects of private life are insulated from these activities. However, 'if the private realm is destroyed, the public is destroyed as well' [Reg95, p. 226]. Hence, without privacy, the self-assurance of citizens to engage in the public could overturn into shallow behaviour with little content. The private space defines the public, and vice versa. It makes people fit into the public space [Sol07].

Earlier in this chapter (see 4.2.2), we argued that privacy provides the context for love, trust and accountability (see [Fri68, Int97, Reg95]). This also applies to the relation between the citizen and the government. For mutual trust and accountability to originate, citizens and public figures need to be provided with a private sphere. Privacy provides a citizen and a public character the environment in which autonomous decisions can be made, in which one is not normalised by others. Subsequently, for these actions individuals need to be trusted and people can be held accountable. However, without privacy, this accountability and trust would not be necessary because such individual decisions are normalised, influenced, and so much transparent that trust would be needless to have.

Protection against state power and state interference is considered to be a core aspect of privacy and relates to the sovereignty and autonomy of people in their 'private environment' [Whi04]. But it is also important to realize that this restraint on the government does not only benefit the individual but actually contributes to government, government figures, and democracy as a whole. State intrusion, e.g., in the form of wiretapping, tracking, and computer cracking, is harmful for democracy and society, which is sometimes overlooked.

4.5   Conclusion

The disadvantages of traditional enterprise centric identity management for both enterprises and individuals are acknowledged and we can observe a move towards federated identity management and even user-centric identity management. The focus in both developments is slightly different. Federated identity management initiatives place enterprise needs at the forefront, at the same time having an eye for the advantages for the individual which lie in the increased convenience these systems provide. User-centric identity management developments place more control and responsibility in the hands of

the individual user, yet also acknowledging the needs of enterprises and governments.

Many of the current developments are still based on the abundant disclosure and collection of personal data to construe rich digital personae. The present chapter has argued that it is important to take privacy seriously. It has done so from three perspectives: an individual perspective, an organisational perspective, and a societal perspective.

At first glance bringing up privacy in a debate about identity management mainly seems to benefit the individual. This chapter has extensively argued that there indeed is an individual interest in privacy protection online in general and in IdM more specifically. The main thrust is that privacy-enhanced IdM allows the individual to play different roles in the online world, just like in the offline world. Being able to separate social contexts and determine how one wants to present oneself to others is an essential individual need to be able to establish and maintain meaningful relations. PE-IdM also empowers individuals to protect themselves and handle the current power imbalance between user and service provider.

Yet, also from an organisational perspective, the domain of enterprises and governments there are clear indicators that privacy needs to play a more important role in IdM. Relations in the online world depend on trust. If consumers and citizens display distrust with respect to their communication partners because these display insufficient attention for privacy and security, this may result in users refraining from using these entities' services. The tide in this respect may be turning judging from experiences with regulation such as the US Security Breach Notification Legislation.

The third level discussed in this chapter is the societal level. We have argued that privacy is a common, public, and collective value that benefits society as a whole. Europeans share a common understanding that privacy matters even though we may disagree to what extent. This warrants treating privacy as a common good. Privacy also resembles a public good such as clean air: we all benefit from its existence and when it is constrained not only individuals but society as a whole will be harmed. Privacy is also comparable to collective goods in the sense that guaranteeing and enforcing privacy on the individual level does not really work.

Society has to take certain actions. One of the actions society can take is enact regulation that guarantees a certain level of privacy protection. This will be the topic of the next chapter: regulating privacy, data protection and identity management.

5

Regulating Identity Management

Eleni Kosta[1], Aleksandra Kuczerawy[1], Ronald Leenes[2], and Jos Dumortier[1]

[1] KU Leuven
[2] Tilburg University

5.1   Introduction

The notions of identity, privacy, personal information and data protection are closely related to each other. Privacy, according to Alan F. Westin 'is the claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others' [Wes67, p.7]. Another definition, provided by Lee Bygrave, states that privacy is 'a condition or state in which a person ... is more or less inaccessible to others, either on the spatial, psychological or informational plane' [Byg02]. Discussions regarding to the nature and sense of 'privacy' is long-lasting and complex. This chapter will not go into this particularly challenging debate, but rather it will sketch the legal framework in which privacy enhancing identity management operates.

   Despite the various understandings of the concept of privacy, it is crucial to keep in mind, what specific interest the law should protect. It is clear that the vital point of a or the 'right to privacy' is the protection against misuse of personal information [Wac, p.10]. As discussed in the previous chapters, the advent of new technologies, have created many new privacy threats, whereas others have just gotten a much wider scope. Some of the already existing risks have changed appearance due to technological advancements. The now famous example of the 'dog poop girl' in Solove's 'The future of reputation' [Sol07] is telling in this respect. The story is about a Korean teenage girl traveling on the subway when her dog pooped. She was asked to clean it up, but refused. In previous times she would have been cursed, but this being the 21st century,

her acts were caught on camera by someone's mobile phone. The pictures were posted on a popular Korean blog. The picture and post went viral and were picked up by the mainstream Korean media. The girl became infamous throughout the country, harassed wherever she went and forced to drop out of university because of the shame. Since the incident, many people, also outside of Korea have seen the images and heard the story.

Privacy-enhancing identity management has a future in limiting privacy threats associated to the online world. However, in order to play such a role and be effective for private and business practices, they have fit into the existing legal framework regarding privacy and data protection. This chapter explores these legal frameworks. The chapter starts by a brief introduction on the European history of data protection regulation in Section 5.2. Next, in Section 5.3, we describe the core principles of the EU data protection regulation. Section 5.4 discusses some of the issues regarding the applicability of the current legal framework in an evolving online world. Finally, Section 5.5 provides some concluding remarks.

5.2 A Brief History of European Data Protection Regulation

The right to privacy protection originates directly from human rights law. The general opinion is that privacy constitutes a fundamental right of the individual and is one of the essential values in a democratic society (see also chapter 4). It can be found in all major international treaties, agreements on human rights and in the constitutions of most countries around the world.[1]

In Europe, one of the first documents recognising the fundamental right to respect privacy was the European Convention of Human Rights and Fundamental Freedoms (ECHR).[2] Article 8 ECHR states that 'everyone has the right to respect for his private and family life, his home and correspondence'. Further, in Article 8(2), ECHR expresses the need to keep the balance between the right for privacy and other interests stating that 'there shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others'. The

---

[1] For an overview of the international instruments in the field of data protection see: http://ec.europa.eu/justice_home/fsj/privacy/instruments/index_en.htm; For an overview of national legislation in over 50 countries see: "An International Survey of Privacy Laws and Developments", Electronic Privacy Information Centre and Privacy International: http://www.privacyinternational.org/survey; See also: http://www.epic.org.

[2] European Convention for the Protection of Human Rights and Fundamental Freedoms (ECHR), Council of Europe, Rome, 1950, http://conventions.coe.int.

lawfulness of these restrictions has been refined in a number of judgements and decisions, issued by the European Court of Human Rights.[3]

Soon after the Convention came into effect it became obvious that the sheer recognition of the fundamental and constitutional principle of privacy is insufficient to effectively safeguard the growing need to protect the right of privacy. This became particularly clear when the full potential of information technologies for controlling data became apparent. This discovery led to a new approach to the issue based on enacting comprehensive national data protection laws applicable to both the private and public sector. Since the start of the seventies many countries followed the trend and enacted more detailed data protection laws. At the same time international developments led to a set of international policy instruments that affected the process of enacting data processing legislation.

The most prominent of these for privacy protection are the Guidelines governing the protection of privacy and transborder flows of personal data issued by the Organisation for economic Co-operation and Development (OECD) and Convention No 108 for the Protection of Individuals with regard to Automatic Processing of Personal Data of the Council of Europe.

The OECD Guidelines, adopted on September 23, 1980, represent international consensus on general guidance concerning the collection and management of personal information. They apply to data held in public and private sector, which pose a threat to privacy and individual liberties, due to the manner in which they are processed, or because of their nature or the context in which they are used. The development of Guidelines aimed to contribute to the harmonisation of national privacy legislation, while complying with human rights, and, simultaneously, to prevent interruptions in international flows of data. This latter aim was considered necessary by the OECD Member countries which feared that disparities in national legislations could hinder the free flow of personal data across frontiers. The guidelines introduce a set of basic principles which should serve as a foundation for national legislations and which should be complied with by the data processors. The principles are: *collection limitation*, *data quality*, *purpose specification*, *use limitation*, *security safeguards*, *openness*, *individual participation*, and *accountability*.

On January 28, 1981, the Council of Europe adopted Convention No. 108. In its preamble, it recognises the need to reconcile the fundamental values of the respect for privacy and the free flow of information between people. It also clearly states that the aspiration of the Council of Europe is to enhance the safeguards for everyone's rights and fundamental freedoms. In particular, the focus of the Council of Europe is placed on the right to the respect for privacy, in order to tackle the new challenges of the increasing flow of

---

[3] Klass,06.09.1978; Sunday Times, 26.04. 1979; Malone, 02.08.1984; Leander, 26.03.1987; Kopp, 25.03.1998; Rotaru, 04.05.2000; Amann, 16.02. 2000; Lambert, 24.08.1998; Valenzuela Contreras, 30.07.1998; Kruslin, 24.04. 1990; Huvig, 20.04. 1990. These judgments are available at: `http://www.echr.coe.int/Hudoc.htm`.

personal data across frontiers and undergoing automatic processing. Just like the OECD Guidelines, Convention 108 spells out a set of principles that should be followed when processing the data. Its main points claim that personal data should be obtained and processed fairly and lawfully; stored for specified and legitimate purposes and not used in a way incompatible with those purposes; adequate, relevant and not excessive in relation to the purposes for which they are stored; accurate and kept up to date; preserved in an identifiable form for no longer than is required for the purpose for which those data are stored; adequately secured; accessible by the data subjects for the rectification or erasure.

Europe, in the mid 1990s, decided to take the lead in harmonizing the data protection regulation. The result of the developments is that current data protection regulation in Europe is primarily based on few key instruments while relevant details specific for particular Member States, their legal systems and traditions, are contained in the national laws in the member states.

5.2.1   The EU Data Protection Directive

The EU went a step further than the OECD guidelines and Convention No 108 of the Council of Europe and enacted regulation for the EU member states pertaining to data protection. The core of data protection is laid down in the general Data Protection Directive 95/46/EC, constituting a data protection framework, and in the Directive 2002/58/EC, known as the ePrivacy Directive, as amended by Directive 2009/136/EC, the Citizen's Rights Directive. Additionally, Directives 2000/31/EC on Electronic Commerce and 1999/93/EC on Electronic Signatures are, to some extent, significant for the current discussion.

The aim of the general Data Protection Directive is to promote the free movement of personal data within the European Union, and to ensure a high level of protection of both, the right to privacy, and of the fundamental rights and freedoms of the individuals with regard to the processing of personal data in all Member States. These two objectives, of ensuring that personal data can move unrestrictedly within the Single Market of the European Union on the one hand, and that a level of protection of the individual's rights on his personal data is uniform within the whole EU on the other, are explicitly mentioned in the Directive's preamble. The fact that the level of protection of privacy provided in national laws of various Member States differed was considered as a major threat to the internal market. It could constitute an impediment to economic activities at Community level, distort competition and impede authorities in the discharge of their responsibilities under Community law. In order to prevent these threats to the internal market, the harmonization of the national laws was desired, with a margin for maneuver left to the Member States. The overall effect of these actions was to result in improvement of privacy protection in the European Community.

The scope of the Directive is very broad as the concept of 'personal data' applies to text, sound and image data. Furthermore, it covers any information relating to an identified or identifiable natural person — a data subject. The Directive clarifies that under the term 'identifiable person' it understands every person who can be identified, either directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity. In order to ascertain whether a person is identifiable, according to Recital 26 of the Directive, account should be taken of all the means likely to be used either by the controller or by any other person to identify the said person. This proves an expansive approach as every data that could be a link to an identifiable individual will come under the scope of the Directive. It brings data, whatever its form, under the 'personal data' umbrella as soon as it is possible to identify the person to whom the information refers, now or in the future.[4] Recital 15 seems to confirm such approach stating that processing of sound and image data is only covered by the Directive, if it is automated or if the data processed are contained in a filing system structured according to specific criteria relating to individuals, so as to permit easy access to the personal data in question.

The concept of 'processing' is defined by the Directive in a similarly broad way. According to Article 2 (b) it refers to any operation performed on personal data such as collection, recording, organisation, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction. This, basically, means any activity that could be performed on data. Even a single consultation or retrieval of a file containing personal data, for example, would constitute processing and has to comply with the provisions of the Directive. Also the sole storage of personal data on a server is considered to be processing, even if nothing is done with the data.

Moreover, the Directive defines several terms relevant for the data subject and introduces specific requirements, which are indispensable in order to render the data processing legal and lawful. These requirements address the 'data controller'. In the context of data protection, 'controller' is every individual or entity who determines the purposes and means of the processing of the data. Who the controller actually is depends on the factual context. In some cases of personal data processing there can be more than one responsible controller. Apart from the concept of data controller, the directive introduced the term of 'data processor', who is a third party who merely processes personal data on behalf of the data controller. The distinction made between 'data controller' and 'data processor' is important for the issue of the liability for violations of the Data Protection legislation. As a rule of thumb, it can be said that the responsible party will be data controller.

---

[4] See also the discussion on whether IP addresses constitute personal data in Section 3.6.6.

In order to prevent the possibility that individuals in the European Union are deprived of any privacy protection if the controller has no establishment in a Member State, the Directive states that it is applicable when the controller makes use of equipment for processing of personal data which is situated on the territory of a Member State. The term 'equipment' covers all possible means like computers, telecommunication devices, impression units, etc. Article 4, however, states an exception to this rule, when the equipment is used only for the purposes of transit of personal data through the territory, such as cables or routing equipment. Moreover, the Directive regulates that if the means for processing personal data are located on the territory of a Member State, a representative established in the aforementioned Member State should be designated by the controller.

The Data Protection Directive, mainly in Article 6, introduces a set of crucial principles for data processing. Most of these conditions refer to the quality of data. These principles set out the core regulation regarding the processing of personal data and therefore they are often characterised as the constitutional law of data protection [Blu02, p.30]. They will be discussed in Section 5.3.

5.2.2 The ePrivacy Directive

The Directive 2002/58/EC, commonly known as ePrivacy Directive, complements the principles introduced in the general Data Protection Directive and converts them into specific rules for the electronic communications sector. The Preamble of the Directive highlights that the advent of new advanced digital technologies in public communications networks in the Community, raises a need for specific requirements concerning the protection of personal data and privacy of the user. The development of the information society automatically leads to the introduction of new electronic communications services and increased access to digital mobile networks by an increasing public. As the capabilities of such digital networks to process personal data are significant, the confidence of users that their privacy will not be at risk is essential for the successful cross-border development of these services. The ePrivacy Directive was modified by Directive 2009/136/EC, commonly known as Citizens' Rights Directive. This Directive introduced the data breach notification and, among others, amended the provisions of the ePrivacy Directive relating to security and confidentiality of personal data, as well as those relating to unsolicited communications. Given that the Citizens' Right Directive was adopted long after the end of the PRIME project, its provisions did not influence the results of the project and will therefore not be analysed at this point.

These risks are especially clear in the area of Location Based Services (LBS). It is clear that in order to enable the transmission of communications, the processing of location data which gives the geographic position of the terminal equipment of the mobile user is required. However, digital mobile networks have the capacity to locate the equipment more precisely than is

necessary for the purpose of transmission of communications. Such accurate data can be used for the provision of value added services such as, for example, providing individualised traffic information and guidance to drivers. In such cases, the Directive states that the consent of the subscriber is indispensable for the processing of such data for value added services to be allowed. Moreover, even after giving their consent, subscribers should be permitted, in a way that would be easy and free of charge, to temporarily or permanently object to the processing of location data. It is also worth mentioning that the Directive emphasises the fact that the protection of the personal data and the privacy of the user of publicly available electronic communications services should be independent of the technology used.

5.2.3 Other Relevant Directives

The main goal of the Directive on Electronic Commerce 2000/31/EC is to regulate the liability of Internet Service Providers (ISPs). All types of illegal activities performed on-line by third parties are covered by the Directive, which adopts a horizontal approach to the issue. This means that it applies to all areas of law, including civil and criminal law. Hence, the liability regulation covers all types of illegal online activities (copyright infringement, unfair competition, misleading advertising, defamation, child pornography, etc.).

Finally, the Directive 1999/93/EC on Community framework for electronic signatures introduced a rule that indicating a pseudonym instead of the signatory's name cannot be prevented by certification service providers who issue certificates or provide other services related to electronic signatures.

5.3 Principles of Data Processing

In this section we will discuss the core principles embedded in Directive 95/46/EC. We will discuss them in the light of defining legal requirements for privacy-enhancing identity management. These requirements can be used as a main guiding tool for the developers of identity management systems and privacy enhancing tools, as was done in the PRIME project. The principles are grouped into three categories: principles on processing of personal data, rights of the data subject and specific requirements for electronic communications systems or applications. Apart from these requirements, we have also defined a set of requirements that are rooted in both law (regulation and legal theory) and in sociology. These latter requirements, i.e., the principle of user consent, principle of security, right to information, right of access and right to rectify, erase or block the data are described in Chapter 6.

5.3.1   Principles on Processing of Personal Data

5.3.1.1   Principle of Fair and Lawful Processing

A fundamental principle laid out in Art. 6(1)(a) Data Protection Directive requires the processing of the data to be fair and lawful. It has been named a primary requirement due to the fact that it 'both embraces and generates the other core principles of data protection laws' [Byg01, p.1]. To assess whether personal data were processed in a fair and lawful way, the method used to obtain the data should be taken into account. Because it is the starting point of processing, it can, to a large extent, influence the fulfillment of other conditions in later stages of processing. In order to have the requirement satisfied, the relevant data subject has to be provided with certain information, mentioned in Article 10 of the Data Protection Directive (on the identity of the controller and of his representative, the purpose of data processing and further information, like who is the recipient of the data, if replies to the question are obligatory or voluntary, and whether there is a right to access and to rectify the data) at the time of the obtaining of the data, or very soon afterwards [Car02, p.54]. Moreover, lawful processing requires the data controllers to comply with all types of their legal obligations, general and specific, statutory and contractual, concerning the processing of the personal data. For example the processing should be performed with respect to Article 8 of the European Convention on Human Rights, which calls for respect for the private life of the individual.

5.3.1.2   Principle of Finality

Article 6(1)(b) of the Data Protection Directive sets the second data processing principle. It is usually addressed under the names of principle of finality, purpose limitation, purpose specification or principle of secondary use. According to this requirement, data controllers must collect data only as far as it is necessary in order to achieve the specified and legitimate purpose. Furthermore, data controllers cannot carry out any further processing which is incompatible with the original purpose. This means that the data subject must be specifically informed about the purpose of the data collection and that subsequent use of collected data is restricted. In particular, the finality principle requires that, without a legitimate reason, personal data may not be used and the concerned individual must remain anonymous. The goal of the principle is to promote transparency and, additionally, to enhance the control of the user over the use of the data. This requirement is seen as the most controversial one in the data protection law [Blu02, p.32]. The indication of the purpose of data collection has to be clear and accurate, using a precise and distinct wording in order to satisfy the principle. This, of course, may lead to a constant dispute over the practical application of the requirement [Blu02, p.32].

5.3.1.3   Principle of Data Minimisation

Article 6(1)(c) of the Data Protection Directive embodies the principle of data minimisation, stating that the processing of personal data should be limited to data that are adequate, relevant and not excessive. The basis for the assessment whether this condition has been fulfilled is the purpose of data collection. Furthermore, Articles 7 and 8 of the Data Protection Directive implicitly repeat the requirement of data minimisation prohibiting the processing of data unless it is indispensable for achieving specific goals. Data controllers are obliged to store only a minimum of data sufficient to run their services. Particularly, data accumulation, a practice often exhibited by public authorities who gather more personal data than required, should be avoided. The storage of large amounts of data can easily be considered as privacy violation, and the argument that the data is not used is insufficient to justify its preservation [Blu02, p.34]. In the context of restrictions on the amount of collected data, issues of 'data avoidance' [HS03] and 'privacy by design' [DG04, p.193] are relevant. The former requires that the technical devices and designs use either no personal data or as limited a amount as possible. The latter suggests that the privacy issues and specifically the processing of personal data (including identity management related implications) should be taken into account from the earliest stage of the organisation of the network infrastructure. Technical tools and Privacy-Enhancing Technologies in particular, should be available to contribute to the effective implementation of the data minimisation requirement.

5.3.1.4   Principle of Data Quality

Another principle, deriving from Article 6(1)(d) of the Data Protection Directive, provides that all personal data shall be accurate and, where necessary, kept up to date. Data controllers are obliged to take every reasonable step to ensure that data which are inaccurate or incomplete, having regard to the purposes for which they were collected are either erased or rectified. This principle is particularly important for the protection of personal integrity. It is often suggested that data controllers should create an appropriate mechanism which would enable the data subjects to update their personal data or notify the data controller about the inaccuracies of the present information. Such solution would prevent, in case of detriment caused by the incorrect data, possible data subjects' complaints of breach of this principle. In practice, these measures are hardly ever implemented.

5.3.1.5   Principle of Conservation

The principle of conservation, also known as the time limitation principle, is described in Article 6(1)(e) of the Data Protection Directive. It stipulates that personal data shall not be kept for longer than is necessary for the purposes

for which these data were collected. It implies that after achieving the purpose for which the data were gathered, they should be rendered anonymous or destroyed, which means that the principle is targeted against the aforementioned practice of data accumulation. It should be emphasised that the processing of personal data for the purpose of anonymisation falls within the scope of the Directive, since the definition of the term 'processing' is so broad that it includes the process of anonymisation as well. However, having in mind the aim of the Directive, imposing compliance obligations with regard to the process of anonymisation could be considered as counter to the achievement of its purpose, especially in light of Recital 26 of the Directive, which says that the principles of data protection shall not apply to data rendered anonymous in such a way that the data subject is no longer identifiable.

5.3.1.6   Principle of Confidentiality

The Directive 2002/58/EC on privacy and electronic communications (ePrivacy) aims to protect the confidentiality of communications. Member States must ensure the confidentiality of communications (and the relevant traffic data) by means of public communications network and publicly available electronic communication services through national legislation. In particular, listening in on, tapping, storing or other kinds of interception or surveillance of communications and the related traffic data by persons other than users, without the consent of the users concerned and except when legally authorised to do so, is prohibited. The Directive provides for an important exception from this principle: legal authorisation for the monitoring of electronic communications is possible when it constitutes a necessary, appropriate and proportionate measure within a democratic society to safeguard national security, defence, public security, and the prevention, investigation, detection and prosecution of criminal offences or of unauthorised use of the communications system (Article 5(1) in conjunction with Article 15(1) of the ePrivacy Directive).

5.3.1.7   Principle of Notification to the Supervisory Authority

The data controller must notify the respective national data protection authority before any data processing operation is carried out (Article 18 of the Data Protection Directive). The Directive leaves to the Member States the possibility to simplify the notification procedure or to waive it altogether in certain situations. However, for the vast majority of entities engaged in any form of automated processing of personal data, the notification remains obligatory. According to Article 19 of the Data Protection Directive notification to a national data protection authority must include at least: the name and address of the controller and of his representative; the purpose of the processing; description of the categories of data subjects and of the data or categories of data relating to them; the recipients or categories of recipients to whom

the data might be disclosed; proposed transfers of data to third countries; a general description allowing a preliminary assessment to be made of the appropriateness of the measures taken pursuant to Article 17 to ensure security of processing.

5.3.1.8 Data Processed in Line with the Rights of the Data Subject

Data controller are obliged to respect the rights of the data subjects when they process personal data. Article 12 of the Data Protection Directive, in particular, grants data subjects the right to be provided, by the data controller, with basic information about the processing of their personal data. It is generally accepted that all the rights mentioned in Article 12 (Subparagraphs (a), (b), and (c)), and not only those from subparagraph (a) as it is explicitly stated in the Directive, should be exercised without constraint at reasonable intervals and without excessive delay or expense [DS97, p.199]. The Directive also provides the data subject with a right to object to the processing of data relating to her (Article 14), as will be elaborated below.

5.3.2 Rights of the Data Subject

The Data Protection Directive grants several rights to the data subjects, although some of them are recognised in an implicit way. Providing the data subjects with those rights intends to guarantee that the data subject remain the ultimate controllers of their personal data. This should also reinforce the fundamental right to privacy described in Article 8 of ECHR. The right to information, the right of access and the right to rectify, erase or block the data will be analysed in detail in the following chapter, as they can be understood as requirements with a social as well as legal basis.

5.3.2.1 Right to Object

Pursuant to Article 14(a) of the Data Protection Directive, Member States shall grant the data subject the right to object to the processing of data relating to him, on compelling legitimate grounds relating to his particular situation. This right to object must at least cover the cases where processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority and where processing is necessary for the purposes of the legitimate interests pursued by the controller (Article 7(e) and (f)).

Article 14(b) of the Directive concerns the processing of personal data for the purposes of direct marketing. The Directive gives the Member States a choice between two formulas. They can grant the data subject the right: (i) to object, on request and free of charge, to the processing of personal data relating to him which the controller anticipates being processed for the purposes

of direct marketing, or (ii) to be informed before personal data are disclosed for the first time to third parties or used on their behalf for the purposes of direct marketing, and to be expressly offered the right to object free of charge to such disclosures or uses. The exact procedure and time limitations to be observed in such cases is the matter of the transposition of the Directive's provisions into national laws.

The right to object is aimed at giving the data subject a possibility to prevent the processing of his data, in case where it violates his personal integrity and where it would be otherwise legitimate. The principle originated from the idea that individuals own their personal data, therefore they should be in a position to control it and oppose to its processing. It is an evident recognition of the right to self-determination.

5.3.2.2 Right Not to Be a Subject to an Automated Decision

Article 15 of the Data Protection Directive grants the data subject a right not to be subjected to an automated decision which produces legal effects concerning him or significantly affects him and which is based solely on automated processing of data intended to evaluate certain personal aspects relating to data subject, such as his performance at work, creditworthiness, reliability, conduct, etc. This right was introduced to overcome the effect of development of information technology which very often leads to decisions being made mechanically. Frequently, such decisions are of essential importance or have legal effects; hence they should be taken by other people who can take into account specific circumstances of the individual. There are statutory exceptions provided to this right in cases where the decision is either taken in the course of the entering into or performance of a contract, provided that the request (for the entering or the performance of the contract) has been lodged by the data subject and there are suitable measures to safeguard the data subjects legitimate interests; or is authorised by a law that also lays down measures to safeguard the data subject's legitimate interests.

5.3.2.3 Right to Seek Legal Relief

Article 22 of the Data Protection Directive provides for a right of every person to a judicial remedy for any breach of the rights guaranteed to him by the national law applicable to the processing in question. Further, the Directive provides that any person who has suffered damage as a result of an unlawful processing operation or of any act incompatible with the national provisions adopted pursuant to the aforementioned Directive is entitled to receive compensation from the controller for the damage suffered (Article 23 of Data Protection Directive).

5.3.3   Specific Requirements for Electronic Communications Systems or Applications

5.3.3.1   Processing of Traffic Data

According to Article 2(b) of the ePrivacy Directive, the term 'traffic data' refers to any data processed for the purpose of the conveyance of a communication on an electronic communications network or for its billing. Traffic data may only be processed to the extent needed for the purpose of the transmission of a communication. When no longer needed for that purpose, the data must be erased or made anonymous (Article 6(1)). Traffic data necessary for subscriber billing and interconnection payments may be processed up to the end of the period during which the bill may lawfully be challenged or payment pursuit (Article 6(2)).

5.3.3.2   Processing of Location Data for the Provision of a Location Based Service

Pursuant to Article 9 ePrivacy Directive, location data may only be processed when they are made anonymous, or with the consent of the users or subscribers to the extent and for the duration necessary for the provision of a value added service. The service provider must inform the users or subscribers, prior to obtaining their consent, about the type of data to be processed, the purposes and duration of the processing and whether the data will be transmitted to a third party for the purpose of providing a value added service. The users/subscribers must also be given the possibility to withdraw their consent for the processing of location data at any time (Article 9(1) of the ePrivacy Directive). It should be emphasised, that location data may only be processed by persons acting under the authority of the provider of the public communication network or publicly available communication services (i.e., the telecommunication operator) or of a third party providing the value added service who obtained the data for the purpose of provision of this service (Article 9(3) of the ePrivacy Directive).

5.3.3.3   Automatic Data Collection Procedures

The data subject has the right to information in case of automatic data collection procedures, as well. Typical examples of such invisible processing include 'browser chattering', automatic hyperlinks to third parties, so-called 'Web-Bugs', active content (e.g., Java) and cookies. Again, the necessary information about the purposes of cookies or similar devices so as to ensure that users are made aware of information being placed on the terminal equipment they are using must be given before any personal data are collected. In particular, the use of cookies (or other tools for storing information on the user's terminal equipment) is only allowed if the user has the opportunity to refuse the cookie to be installed. However, this condition does not apply if the use of the cookie is

"strictly necessary in order to provide an information society service explicitly requested by the subscriber or user" (Article 5(3) of the ePrivacy Directive).

5.3.3.4   Unsolicited Commercial Communications (Spam)

The ePrivacy Directive is also an important step forward in the protection of the users of electronic communications against unsolicited messages. The use of automated calling systems without human intervention (automatic calling machines), facsimile machines (fax) or electronic mail for the purposes of direct marketing may only be allowed in respect of subscribers who have given their prior consent (opt-in). As an exception to this general rule, it remains possible for merchants to send electronic mail to their own customers for the purpose of direct marketing of similar products or services, provided that customers clearly and distinctly are given the opportunity to object (opt-out). Other types of unsolicited communications for purposes of direct marketing are not allowed either without the consent of the subscribers' concerned (opt-in), or in respect of the subscribers who do not wish to receive these communications (opt-out). In any event, the practice of sending electronic mail for purposes of direct marketing disguising or concealing the identity of the sender on whose behalf the communication is made, or without a valid address to which the recipient may send a request that such communications cease, has to be prohibited by Member States' legislation.

5.4   Applicability Issues of the Current Legal Framework

5.4.1   An Old Directive for New Technologies

The principles included in the general Data Protection Directive, as well as their specific interpretation in the ePrivacy Directive in cases where data protection issues arise in connection to publicly available electronic communications services and networks, delineate a solid data protection framework at the European level. At first and overall glance, the European legal framework on data protection contains the core principles that can ensure the protection of individuals with regard to the processing of personal data on one hand and the free movement of such data on the other. These were the main objectives of the Data Protection Directive back in 1995 and it can not be contested that they actually still ensure a satisfactory level of protection of the individuals when the processing of their personal data takes place in a conventional way, for instance when data are collected and processed by a company, with whom the individual signs a contract.

Objections regarding the effectiveness of the Directive arise with regard to new technologies. As already illustrated in Chapter 3, the notion of personal data is not always clear when new technologies are involved. IP addresses,

cookies, RFID technology are only but a few examples that show that the application of the Data Protection Directive is not free of problems. There is just too much information, created and exchanged in too many different ways. A piece of information, which relates to an identifiable natural person under one circumstance, does not qualify as personal data in another situation. Although the Directive was written up in a technologically neutral way, some new developments reveal the vulnerability of the Directive to deal with them efficiently. The European Commission actually admitted in its Communication on the follow-up of the Work Programme for better implementation of the Data Protection Directive that "the extensive development of new information and communication technologies necessitates specific guidance on how to apply [the] principles [laid down in the data protection directive] in practice" [otEC07, p.10].

Does this mean that a completely new piece of European legislation is needed? As the European Data Protection Supervisor has articulated, "there is no need for new principles, but there is a clear need for other administrative arrangements, which are on the one hand effective and appropriate to a networked society and on the other hand minimize administrative costs" [EDP07, p.4]. In simple words, this would mean that the most important principles for data protection are laid down in the Directive, so there is no pressing need for a new piece of legislation. Although new developing technologies reveal the vulnerabilities of the current legal framework, it is technology that can give the solution to this problem, when "used effectively and [is] relied upon in a privacy enhancing way" [EDP07, p.6]. It is the relation between technology and law that needs to be redefined: law enabling technologies and technologies enabling the law are the only solution that can ensure adequate protection of the individuals, when processing of their personal data is involved (see also extensively on the interplay between law and technology in this respect [Han08, Lee08, KL05]).

5.4.2 The Role of the ePrivacy Directive with Regard to the Challenges Posed by New Technologies

The general Data Protection Directive is complemented by the ePrivacy Directive, when processing of personal data in the electronic communications sector is involved. The ePrivacy Directive aimed at the protection of the users of publicly available electronic communications services that are offered via public communications networks regardless of the technologies used, seeking to implement the principle of technology neutrality into the regulation of data protection in the electronic communications sector (Recital 4 of the ePrivacy Directive). However, questions arise regarding the applicability of the ePrivacy directive to several emerging technologies, such as RFID, and to problems that arise from their use in the field of electronic communications.

Although the distinction between private and public networks seemed reasonable at the time of the drafting of the ePrivacy Directive, the fact that the

Directive only applies to publicly available electronic communications services in public communications networks is heavily criticised today. The Article 29 Working Party on Data Protection has expressed the opinion that "private networks are gaining an increasing importance in everyday life, with risks increasing accordingly [and there is a] tendency [that they] increasingly become a mixture of private and public ones" [Par06, p.3]. The same opinion is shared by the European Data Protection Supervisor, who "regrets that the proposal [for a Directive amending, among others, the ePrivacy Directive] has not tackled the issues of the increasingly blurred distinction between private and public networks" [EDP08, p.6].

Nevertheless, it seems that the ePrivacy Directive will still apply only on public networks and services, even after the review. It shall be clarified that the individuals enjoy the protection of the general Data Protection Directive, whenever processing of personal data takes place. It remains to be examined whether the specific provisions of the ePrivacy Directive that regulate issues, such as security, confidentiality, traffic and location data, are also applicable. Currently in order to decide upon the applicability of the ePrivacy Directive, three main issues need to be examined:

1. Whether there is an *electronic communications service*,
2. Whether this service is offered in a *communications network* and
3. Whether the aforementioned service and network are *public*.

According to Article 2(d) of the Framework Directive[5] "*public communications network* means an electronic communications network[6] used wholly or mainly for the provision of publicly available electronic communications services[7]". The term *communication* is defined in Article 2(d) of the ePrivacy

---

[5] Directive 2002/21/EC of the European Parliament and of the Council on a common regulatory framework for electronic communications networks and services (Framework Directive), O.J. L 108, 24.04.2002, pp. 33 - 50.

[6] '*Electronic communications network* means transmission systems and, where applicable, switching or routing equipment and other resources which permit the conveyance of signals by wire, by radio, by optical or by other electromagnetic means, including satellite networks, fixed (circuit- and packet-switched, including Internet) and mobile terrestrial networks, electricity cable systems, to the extent that they are used for the purpose of transmitting signals, networks used for radio and television broadcasting, and cable television networks, irrespective of the type of information conveyed' (Art. 2 (a) Framework Directive).

[7] '*Electronic communications service* means a service normally provided for remuneration which consists wholly or mainly in the conveyance of signals on electronic communications networks, including telecommunications services and transmission services in networks used for broadcasting, but exclude services providing, or exercising editorial control over, content transmitted using electronic communications networks and services; it does not include information society services, as defined in Article 1 of the Directive 98/34/EC, which do not consist wholly or mainly in the conveyance of signals on electronic communications networks' (Article 2 (c) Framework Directive).

Directive as "any information exchanged or conveyed between a finite number of parties by means of a publicly available electronic communications service. This does not include any information conveyed as part of a broadcasting service to the public over an electronic communications network except to the extent that the information can be related to the identifiable subscriber or user receiving the information".

The need for further clarification of these quite complicated definitions has already been recognised by the Article 29 Working Party: "The Working Party notes that both definitions 'electronic communications services', and 'to provide an electronic communications network' are still not very clear and both terms should be explained in more details in order to allow for a clear and unambiguous interpretation by data controllers and users alike" [Par06].

5.5   Conclusion

The European data protection framework tries to strike a balance between promoting the free movement of personal data within the European Union, and ensuring a high level of protection of the right to privacy, and of the fundamental rights and freedoms of the individuals with regard to the processing of personal data in all Member States. This means that the Directive promotes the free flow of information provided that a set of data protection principles is observed. The basic data protection principles for the processing of personal data contained in the Data Protection Directive provide a certain level of protection. The provisions in the Directive (through their implementation in the legislation of the member states) provide obligations for data controllers and rights for data subjects and should be observed in the implementation of any data processing system that deals with (potential) personal data. The principles outlined in this chapter are therefore also design requirements for privacy-enhancing identity management solutions.

The protection seemed adequate at the time the Directive was written. The tide, however, seems to shift. The development of new technologies and new services create new challenges with respect to privacy and data protection. The basic data protection principles need to be revisited in order to be able to tackle the challenges of today. This does not necessarily need to be done by a new legislation. The solution to upcoming challenges may be provided by what causes them in the first place: technology. Technology may provide solutions that will enable the privacy compliant processing of personal data. PETs can play an important role in implementing and enforcing the data protection principles. Data minimisation, anonymisation and purpose limitation are just three of the principles that can be realized in privacy-enhancing systems as we will see later on in this volume.

6

User-Centric Privacy-Enhancing Identity Management

Bart Priem[1], Eleni Kosta[2], Aleksandra Kuczerawy[2], Jos Dumortier[2], and Ronald Leenes[1]

[1] Tilburg University
[2] KU Leuven

6.1 Introduction

Online identities are associated to individuals and improper handling of these identities may therefore affect these individuals. Placing the individual at the center of identity management and empowering them with tools to actively manage their identity may help limit the privacy risks provoked by the information society. As we have argued in the previous chapters, embedding privacy into the design of identity management systems is important. What the actual embodiment of privacy features into IdM encompasses is less clear. The previous chapter has shown a number of data protection principles that have to be observed by any system that handles personal data. These principles are part of the legal requirements for the development of any application that handles personal data, including identity management systems. There are also other sources of requirements. Human computer interaction research, sociological research and economics/business studies can also contribute to defining requirements for privacy-enhancing identity management systems. In the current chapter we focus on results obtained in PRIME research in the fields of law and sociology and human computer interaction that resulted in a set of concrete set of requirements for user-centric privacy-enhancing IdM. A more detailed description of user-focused privacy requirements can be found in PRIME's Deliverables Framework V3 [PRI08] and Requirements V3 [KDR$^+$08].

Section 6.2 briefly discusses the sources of the requirements described in the current chapter. Section 6.2.1 deals with the importance of *audience segregation* in Identity Management, and its direct link with privacy. One

important aspect of audience segregation is *user control*. User control, a complex and ambiguous concept that gives rise to a set of subrequirements, is addressed in detail in Section 6.2.2. These requirements stem from legal and sociological/psychological grounds. Section 6.2.3 concludes the chapter by discussing a number of *adoption* requirements that should guarantee the user adoption of privacy-enhanced identity management developed along the lines of the previous requirements.

6.2 Sources of the User-Perspective Requirements

Legal and sociological research within the PRIME project has contributed to the conception of a set of requirements for privacy-enhancing identity management from a user-perspective. Identity management systems must comply with data protection legislation. The legal data protection principles outlined in the previous chapter are obvious starting points for developing requirements that do justice to the user-perspective of identity management systems. The current legal privacy-framework was therefore analysed in chapter 5 from the perspective of the individual as a user of identity management systems. The relevant Directives are:

- Directive 95/46/EC (Data Protection Directive),
- Directive 2002/58/EC (ePrivacy Directive) and,
- Directive 2006/24/EC (Data Retention Directive).

Apart from those, also the European Directive 1999/93/EC (Electronic Signature Directive) and the European Directive 2000/31/EC (eCommerce Directive).

The legal framework provides some general requirements for privacy-enhanced IdM systems. Another source for user-perspective requirements is literature on social aspects of interaction and technology use and privacy literature in general, when viewed through the lens of the individual. The input to the 'social' requirements comes from sociology, HCI, eCommerce, marketing, law, and philosophy research (e.g., [JB05, PK03]). Also survey data relating to privacy and identity management was was incorporated in the process of deriving requirements.[1]

6.2.1 Audience Segregation

Audience segregation is an essential aspect of Identity Management for the individual (see also Chapter 4). Every individual has different characters, which are used in different settings in society, such as 'citizen', 'daughter', 'friend',

---

[1] The survey results obtained in a large scale survey conducted within the PRIME project under Dutch, Flemish and UK students can be found as an annex to PRIME deliverable Requirements V3 [KDR+08](version 2.0 May, 2008), which is available from the PRIME website http://www.prime-project.eu.

and 'employee'. In playing their characters (which are sometimes roles), people explicitly and implicitly disclose information about themselves. This information people give, and give off [Gof59], is determinative for their character. While deploying or combining information, individuals are to some extent able to construct and manage their different characters in life, facilitating them to have various relations with different levels of intimacy. However, to be able to play different characters, one needs to be able to control the attributes of these characters and the settings in which they appear.

Audience segregation is an issue in the online environment, because 'simple' partial identities (or digital personae [Cla94]) can be aggregated into rich compound identities from data linked to identifiers, such as names and IP-addresses. Digital personae are easily copied, merged and manipulated. Hence, digital personae can be exposed to 'audiences' that should not be able to see them and be able to obtain personal data. This is even possible without the individual being aware of its occurrence. The merging of data and use of data out of context can easily result in practices such as social sorting and discrimination. A lack in the ability to segregate audiences also increases the risks of reputation damage because critique, comments, and worse online bullying, or blackmailing, for example, easily cross audiences.[2]

Having different partial identities is a social necessity. It allows the individual to fit into different social spaces, like work and family. Characters are furthermore often required to 'team play' in relations with others, like family and colleagues. Having consistent characters and segregating audiences positively affects the relation with relations present in a specific social context. In addition, characters are important in the sense that being confronted with the individual out of character may lead to wrong interpretations of behaviour, confusion, and decisions based on 'wrong' (out of context) information. For instance, bringing up certain hobbies in a job interview, may turn out not to be a good idea. The fact that one keeps snakes and feeds them mice, may not have a positive impression on the person conducting the job interview, while the hobby may well not at all affect the professional performance of the candidate.

The necessity to segregate audiences and play characters is an essential aspect of informational privacy. Having a variety of relations, or being able to develop oneself, is not only determined by the information we share in relations, but also by the information that is (mutually) concealed [Sch68]. In addition, not knowing something about a character or not needing to know information directly relates to the notions of trust, autonomy, cohesion, efficiency, and accountability (see, e.g., [Int97, Fri68])

If identities become 'mixed up' segregating performances played in different relations and relations is no longer possible and relations run the risk of becoming one-dimensional, confusing, and shallow. Lack of audience

---

[2] As Solove's [Sol07] 'Dog poop girl' example shows. See Chapter 5 for the details.

segregation would make an individual the same to his employer, spouse, dentist, best friend, and parents: everyone would become one-dimensional and colourless.

Some privacy concerns voiced by users in privacy studies clearly relate to this dimension. Many students in the PRIME survey, for instance, state that they use different and anonymous e-mail addresses to separate contexts (business, social) (see [KDR+08] May 2008 version). One of the key requirements that can be derived from the need for being able to segregate audiences is user control.

6.2.2   User Control

Even though there are many privacy conceptions, user control in many is a core requirement [Fri68, Rac75, Wes67]. User control ranges from some influence on what gets disclosed to whom, up to very strong positions such as the German right to informational self-determination. Both user control and self determination are part of the European notion of privacy [Sta02, OMS+07, PRI06a], and acknowledged in national and European data protection regulation. User control is therefore also a key requirement for privacy-enhanced IdM systems. Control, however, is an ambiguous concept [Gav80] which therefore needs to be explored into more detail. The following sections decompose user control into manageable concepts and preconditions for ex-ante and ex-post user control. We do this, from a social and legal point of view. We distinguish five sub-requirements: *information to the user*, *consent of the user*, *user access*, *correction, erasure, and objection*, and *security and trust*.

6.2.2.1   Information to the User

In order to be compliant with Article 10 of the Data Protection Directive (95/46/EC), a data controller should provide a data subject some minimum information regarding the processing and the controller doing the processing (cf. Chapter 5). A privacy-enhanced Identity Management system needs to take this obligation into account. Providing information to the user is an interpretation of the legal principle of fair and lawful processing because only when a user is informed beforehand about data collection, he or she can assess a service and decide whether or not to participate. In addition, providing information prior to the disclosure increases the willingness of people to enter into a relationship, a step in creating the social contract between data subject and data controller. It is therefore also a precondition for users to know when they can exercise their rights. Providing information to a user therefore is the first and crucial step to empower the individual to construct and maintain their identity and guard their privacy.

According to the Data Protection Directive, the minimum information that needs to be provided to the user, concerns:

1. The identity of the controller or his representative;
2. The purposes of the processing for which the data are intended;
3. Any further information if this is necessary to guarantee fair processing in respect of the data subject, such as the recipients or categories of recipients of the data, whether replies to the questions are obligatory or voluntary, as well as the possible consequences of the failure to reply and the existence of the right of access to and the right to rectify the data concerning her.

The information has to be provided to the user at the time — or before — their personal data are collected. If disclosure to a third party is foreseen, Article 11 of the DPD provides that the information must be provided at the latest when the personal data will be disclosed to this third party. The Directive excludes the right of information in cases where the disclosure to a third party is made for statistical purposes, or for the purposes of historical or scientific research, and when 'the provision of such information proves impossible or would involve a disproportionate effort or if recording or disclosure is expressly laid down by [national] law' (Article 11(2) of the Data Protection Directive).

Information is a key prerequisite to providing the user control over their personal data. Data subjects need to know what will happen to their data and indirectly to themselves. This promotes their autonomy and fosters human dignity. Having information at their disposal also raises the 'consciousness' of the data subject, which is essential to enable them to make informed choices concerning the dissemination of their personal data. Moreover, when information regarding data collection is provided, the process of data collection is made transparent beforehand, which contributes to fairness and trust. In addition, information about processes of data collection reduces the chances of instituting 'panoptic surveillance', in which human behaviour becomes normalised and influenced by the sense of omnipresent surveillance [Fou77].

Being aware which data will be collected and for which purposes may reduce the risks that the data controller can collect data to serve as a basis for many — potentially undesirable — processes and decisions, like profiling, discrimination and exclusion. The transparency this creates is an instruments that helps level out the immanent power-imbalance between data subject and data controller.

The information that is given to the user is seen both as a right of the data subject and as an obligation of the data controller to inform the data subject. In practice, the obligation to inform the data subject is seen as a major duty of the data controller, as the data subject very often is ignorant of the fact that processing of some of her data takes place, let alone knows the details regarding the processing. Only providing the user a minimal right to information will probably not guarantee the actual consciousness of a data subject to the data processing and its effects. It is therefore necessary to go beyond providing the minimal information and also raise awareness regarding the essential events, stakeholders, and attributes of the collection and use of personal data. This requires that information is presented in a comprehensive format. This is a

difficult task because of the different information needs of people and their capacity to understand the information. 'Comprehension' of the information provided is essential because only then can misguided disclosure of information, false information sharing, and user regret be minimized. This makes the provision of clear information not only an the interest of the user, but also of the data controller because it avoids future conflicts or unsatisfied customers.

Following from the requirement to provide information in a way that creates *consciousness* and *comprehension* is that information needs to give users a glance into the future. Privacy-enhanced Identity Management systems therefore need to be *consistent*. Many to all actions following from the collection of data lie in the future, and so there is always a risk of future misrepresentation of partial identities or unforeseen and unwanted decisions. People, preferences, and situations change and data may be used differently in the future. By providing the user a consistent application, however, a level of trust is integrated, and can people anticipate to the future use of their personal data. If consistency is not taken into consideration, there is a risk that things 'go weird' which can damage the perceived trustworthiness of an application. Showing the normal line of operation to a data subject makes it possible for users to estimate the future consequences of their actions. In addition, providing the user complementary information, e.g., in the form of markers, warranties, and seals can contribute to the trust of a data subject in data transaction parties.

6.2.2.2   User Consent

Legitimacy of data processing according to the principle of legitimate data processing, requires the unambiguous consent of the data subject. Consent is of major importance, because it changes an unlawful act info a lawful one. In this sense, consenting to data processing makes the difference between an infringement on privacy or an allowed use of personal data [Wes04].

Consent should be voluntary and in most of the cases shall be revocable. Moreover, influences of force, fraud, incompetence, and paternalism need to be rejected. In this respect, hierarchical relations deserve special attention. Because consent of a data subject can be influenced and manipulated by many factors, the Data Protection Directive (95/46/EC) stipulates that the data subject's consent shall mean any 'freely given specific and informed indication of her wishes by which the data subject signifies her agreement to personal data relating to her being processed' (Article 2(h) data protection directive).

It is very important for the data controllers to interpret the aforementioned legal provision correctly in order to avoid violations of the data protection legislation. An important issue is what 'freely given, specific and informed' means. Freely given consent shouldn't be conditional on an advantage or subject to negotiations on behalf of the data controller. The consent needs to be specific, meaning that it should be given for a specific and identified scope. Finally, it needs to be informed; the user shall get the appropriate and sufficient

information before the collection of the data and such information shall be in clear language and of course in a language that the data subject understands. In this last demand, we can see the relation with the requirement of consent with the previous requirement, 'information to the user'.

A highly debated issue is whether consent can be expressed in an opt-in or in an opt-out way. It is necessary that 'there must be some form of communication whereby the individual knowingly indicates consent'. This can be expressed by ticking a box, or sending an e-mail or subscribing to a service. For the processing of sensitive data, i.e., data that reveal racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership or data concerning health or sex life, the data subject shall give her explicit consent, although Member States may even prohibit the processing of sensitive data, even with the consent of the data subject.

It shall furthermore be noted that the definition of consent explicitly rules out consent being given as part of accepting the general terms and conditions for an offered electronic communications service. Many contemporary services disregard this requirement. In current practice consent is usually obtained through the general terms and conditions of a service offering (in which the processing of personal data will occur). The picture gets even blurrier when the consent of the user is given in an environment that allows no or minimal user interface, such as in the case of most emerging technologies, like RFID, Bluetooth, etc.. Ambient Intelligent environments are based on the processing of personal data, and obtaining the consent of the data subject is often not taken into account in the designing phase of these systems.

Related to the requirement of consent is *choice* as an important social condition for true privacy-enhanced Identity Management, because consent implies a possibility for the user to choose whether or not to engage in a service and subsequently to choose how her privacy requirements are addressed in different services. When service providers use a 'take-it-or-leave-it' approach (viz. without offering different privacy options), it is impossible for users to withhold specific information from the focused attention of others. Individuals need to be enabled to choose for themselves the way they are portrayed to others, instead of being bound to predetermined identities and predetermined judgments. However, for the sake of motivation and feasibility, choice should not be exaggerated, but moderated and limited [Hey02].

Next to choice and consent, individuals also need to be able to set the boundaries in which their data is being used. Such *'confinement'* [JB05] relates to the purpose of use of data, but also to security measures. Data controllers may define the purpose of use and access to data too broadly or incomprehensively for the user resulting in an undermining of their privacy position. The user should therefore be enabled to define purpose of use and access to data, to avoid data leakage to others and/or to circumvent the use of data for unintended purposes ('function creep').

6.2.2.3   The Users' Right to Access the Data

User control would be a useless concept if individuals are unable to inspect whether actions with regard to data collection observe their policies. Ex-ante control is insufficient to ensure privacy. Moreover, data can be interpreted or presented wrongfully, users can make mistakes, change their preferences, or regret earlier decisions. Access to disclosed data is therefore necessary to enable users to check whether data controllers observe the agreements with them, observe the legal requirements, and to assess whether the data collected and processed is correct. Users should also be able to inform data controllers about possible errors or harmful behaviour by them. Just like 'information to the user' contributes to ex-ante transparency, the right to access data contributes to ex-post transparency and helps level the asymmetric power relation between data subject and data controller. The requirement of access to data furthermore relates to the general legal principle of data quality, because it allows users to notice and correct wrong personal data.

The Data Protection Directive grants various rights to data subjects with regard to the processing of personal data. The right of access to collected personal data states that every individual of which personal data has been collected and processed has the right to obtain from the data controller:

- confirmation as to whether or not her personal data are being processed and information at least as to the purposes of the processing, the categories of the data concerned, and the recipients or categories of recipients to whom the data are disclosed,
- communication to her in an intelligent form of the data undergoing processing and of any available information to the resources and of any available information as to their source.

Where any automated decisions (as defined in Article 15 of the Data Protection Directive) are involved, the data subject has the additional right to be informed about the logic involved in any automatic processing of data concerning her. All the aforementioned information must be available to the data subject 'without constraint at reasonable intervals and without excessive delay or expense' (Article 12 (a) Data Protection Directive). In addition and as regards to how the right of access is exercised, an ideal situation would include both online and physical access — the latter realised at the physical address of the data controller. However, in cases where physical access would entail disproportionate efforts and costs on behalf of the data controller (or if the data collected is disproportionately little), it is arguably accepted that the right of access can be exercised only through online means. In such a case however, the controller shall ensure via strong authentication mechanisms that the person requesting some information about the processing of personal data is the one entitled to do so, in order to avoid cases of identity fraud, identity theft etc.

As already mentioned, access and inspection contribute to the fairness of data processing and decreases the power imbalance between the strong party

(data controller) and the weak party (data sharer) in a data collection process. Access and inspection are thus countervailing powers. These powers should not only be applicable to the initial data collector, but throughout the *chain* in which a service is being delivered to the user. Services are often provided by combining the efforts of several organisations. The telecommunications sector for instance, has multiple parties engaged in the provision of one single service (see for instance Chapter 25). Furthermore, business processes and the data processing involved can be outsourced to other (specialized) parties. Users should therefore not only have insight in the phase of initial data collection, but also in phases such as subscription, payment, and integration of a service.

6.2.2.4   Rectification, Erasure, and Blocking of Data and the Right to Object

People can make mistakes or regret their decision concerning the dissemination of personal information. Initially, one can be tempted to disseminate personal information, as the benefits of personal data disclosure are much clearer than their disadvantages [Sta02]. Negative effects of data disclosure may occur later in time when people encounter undesired use of their data or even downright abuse of personal data. Apart from this reason to grant a right to withdraw data, people need to have the ability to decide to continue or modify their behaviour when their lives change or when personal data turns out to be wrong or interpreted incorrectly. IdM systems need to provide a level of 'forgetfulness' which is not present by default in the online environment [BJ02].

The 'right of access' to one's own personal data in the broad sense includes a right to rectify, erase, or block the data that relate to the data subject in cases where the processing does not comply with the requirements of the Data Protection Directive. For example, the data controller's collection of personal data may turn out to be disproportionate to the purposes, or when the data at issue are incomplete or inaccurate (Article 12 (b) of the Data Protection Directive). A common instance where data subjects exercise their right to rectify data is when their name is misspelled and they ask for correction. Furthermore, in the course of ex-post control over their personal data, the data subject also has the right to object (Article 14 and Recital 45 of the Data Protection Directive) to the collection and processing of her personal data. These aforementioned rights can only be imposed upon the data controller when the data subject has a legitimate right to do so and at the data controller does not have an overriding right to process the data. It is important to note that consent of the data subject is only one of different reasons according to which the processing of personal data can take place, so the right to object can not for instance be exercised in front of a data controller who deems that the processing is necessary for the performance of a contract to which the data subject is party.

Nevertheless, Article 14 of the Data Protection Directive stipulates the cases where the right to object can be exerted. Firstly, when the processing is

necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the data controller or in a third party to whom the data are disclosed and when the processing is necessary for the purposes of legitimate interests pursued by the data controller or by the third party or parties to whom the data are disclosed, except where such interests are overridden by the interests for fundamental rights for fundamental rights and freedoms of the data subject, Member States are obliged to grant the data subject a right to object at any time on compelling legitimate grounds relating to her particular situation to the processing of data relating to her, save where otherwise provided by national legislation. When there is a justified objection, then the processing instigated by the controller may no longer involve those data. Secondly, the data subject can object, on request and free of charge, to the processing of personal data relating to her which the controller anticipates being processed for the purposes of direct marketing, or to be informed before personal data are disclosed for the first time to third parties or used on their behalf for the purposes of direct marketing, and to be expressly offered the right to object free of charge to such disclosures or uses.

The ePrivacy directive perceives the right to object and the right of withdrawal of consent in various situations. Therefore, the specific right is implicitly mentioned as a right to object to the installation of cookies, to the processing of traffic data processed for the purpose of marketing electronic communications services or for the provision of value added services, the processing of location data other than traffic data, to have her data available in directories of subscribers and to the processing of her personal contact information in order to receive unsolicited communications. In all the aforementioned cases, the data subject is given the right to refuse the provision of services or in cases where she has already accepted them, to withdraw her consent.

The requirement of 'access to information' would lose its value if subsequent actions cannot follow from this inspection of information. Thus, ex post user control by erasing, blocking, and correcting information is closely related to, and follows from, access and inspection. This requirement can serve the social need for 'forgetfulness', when people feel the need to get a 'fresh start' or 'second chance' in life [BJ02]. Moreover, a world in which people cannot make mistakes and nothing is forgotten is not a world conducive to the development of democratic and autonomous individuals. There of course is also a need to hold users accountable for their behaviour and the information they share which has to be balanced with data erasure. Also we have to take into account that the responsibility for the quality of data lies at the data controller. Because of this, ex-post user control by blocking, erasing, and rectifying information, is a balancing act between what is (legally) necessary to achieve accountability of the user, correctness of data, and the (legal) possibility to provide the user a control tool which can complement the data controllers' obligation with regard to the quality of data.

6.2.2.5   Data Security and Trust

An important prerequisite for user control is a secure infrastructure because if, for instance, third parties have access to the communication between data subject and data controller or to the collected data, user control is relatively meaningless. Therefore the data controller needs to take appropriate security measures. From a social perspective, the need for security is also related to trust, which is a highly relevant aspect for the success of online transactions. Even though trustworthiness and security are not the same, many users will not be skilled to asses the security measures taken by a data controller and therefore have to rely on trust marks provided, for instance, by institutions they do trust. Which institutions are trusted by individuals depends among other factors on context and culture.

Data security requires data controllers to take 'appropriate technical and organizational measures' (Article 17 (1) of the Data Protection Directive) against unauthorised or unlawful processing, and accidental loss, destruction or damage to the data. To the extent that this principle covers the security requirements and robustness of the network itself, this principle overlaps with the security and confidentiality requirements laid down in Articles 4 and 5 of the ePrivacy Directive (Directive 2002/58/EC). Taken as a whole, this principle imposes a statutory obligation on data controllers to ensure that personal data are processed in a secure environment. This means that the data controllers must consider the state of technological development and the cost of the implementation of any security measures.

Bearing in mind these factors, the security measures that are adopted by the data controllers must ensure a level of security that is appropriate to both the nature of data to be protected and the likely harm that would result from a breach of this principle. It follows that, the more sensitive the data, the more adverse the consequences of a security breach would be for the data subject, and therefore more stringent security requirements should be put in place.[3] This is especially the case as regards the processing of health related data. In any case, the data controllers should implement appropriate security measures to ensure that non-authorised personnel are unable to gain access to personal data. In addition, security precautions would suggest making secure back-up copies.

Security measures are of importance to ensure that boundaries for data processing determined by the user, are not crossed. Without appropriate security measures, confinement of data processing is thus not possible. Another relevant aspect when discussing security is that infrastructure and transaction partners need to be trustworthy. Security and or security marks can play a role in increasing and signaling trustworthiness. Not only should an organisation thus handle a secure transaction of data, they should also make these risks and their measures transparent to the user. The user needs to recognize the security and reliability of a technology and the trustworthiness of an

---

[3] See on this aspect also Section 7.3.

organisation, which is difficult to achieve because many users are laymen in the field of technology and security, and online transactions lack face-to-face interaction.

Trust is commonly conceived of as the assumption that another person, organisation, and its technology will not take advantage of the vulnerable party. Turned around, a trustworthy data controller should be trusted to attend to the interests of the data subject. By its very nature and by the differences in social context, trust is defined differently amongst social groups and individuals. However, some generally regarded trust marks — like trust seals — can contribute to the trustworthiness of an application and the organisation that uses the application.[4] These markers may originate from well-reputed organisations, and should not only apply to the specific security measures (which for most users are difficult to comprehend), but also to information about sources, providers, affiliations, and certificates of the data processor. A broad use of markers is necessary, whereas there is a general problem with regard to trust in technology: the information about security and trustworthiness needs to be tailored to the context of the (non-expert) user.

Trust in technology will often be combined with the trust in the partners one engages with. This is also important considering the adoption and use of a privacy-enhanced service. For the sake of trust and the adoption of a technology, complementary markers about reputation and brand of a data controller/service provider can therefore also be of importance.

6.2.3 Adoption of Privacy-Enhanced IdM in Society

Privacy is pursued in a specific social environment and has social importance, which effects the adoption of privacy-enhanced technologies. In addition, the adoption of privacy enhancing technologies relies on general aspects of technology adoption, like product aspects and market factors. Some of these market factors will be described in the next chapter. Some social aspects regarding the adoption of PETs will be elaborated in this section.

There is a plethora of privacy-enhancing technologies available on the market (some freely available), but adoption of these technologies by the individual seems to be difficult, even though people generally are concerned about their privacy in online environments [Sta02, BGS05, Sho03]. This demonstrates that adoption of a privacy-enhanced IdM system is not obvious. Given the importance of privacy for collective, common, and public values, broad adoption of PETs is desirable. Broad adoption, instead of use by only a few users, is also necessary in order to prevent 'digital divides', or 'digital inequality' regarding privacy protection online [DH01] and to create a multiplier effect. Thus, the ability to access and use technologies needs to be guaranteed for every online user in order to limit digital inequality in societies. In this respect, two aspects are important: *affordability* and *skill level*.

---

[4] An initiative to provide comprehensive privacy trustmarks is the EuroPrise privacy seal, see https://www.european-privacy-seal.eu/.

6.2.3.1   Affordability and Skill Level

The first requirement is affordability of privacy enhancing IdM solutions to a large group of users. There is a widespread notion that people are, at the moment, unprepared to pay much for privacy [Sho03]. There is no consumer market for privacy, because the benefits of ensuring privacy on an individual level do not seem to be clear and are difficult to define economically, whereas the benefits for giving up privacy are clear and often bring direct advantages [Sta02]. In this sense, affordability is related to the perceived usefulness of a privacy-enhanced IdM system. Information about the product and comprehensibility of its features can therefore influence the perceived affordability of a service. Currently, there does not seem to be a high level of the necessity of PETs amongst individuals, although privacy-awareness will probably be increasing when technologies become a part of our everyday life.

In addition, users should be able to use an application with a minimal amount of training. Not only actual access to the technology, but also skills and motivation can determine equality in society. Groups with relative low skill levels, like children or the elderly also make use of the online environment, and should also be able to protect their privacy. There is no homogeneous user group, and skills can even change between social groups or nations. Because of this, it is necessary that IdM systems can be used by non-skilled users and provide satisfactory default privacy settings.

6.2.3.2   Context and Social Settings

People value privacy differently. Some of us are 'privacy fundamentalists', whilst others may share personal data without hesitation. On top of this, situational factors add complexity, because the use of identities and identity-related information is adjusted to the environment and the kind of interaction people engage in. Thus, information that is considered private changes throughout situations. One can for example relate to the difference between sharing information at a crowded helpdesk or at a birthday party, or to the difference between disseminating personal data to authorities or best friends. Moreover, sharing medical data with a doctor may not be considered privacy sensitive, but sharing the same data with a real estate agent may be completely different. These examples illustrate that it thus is difficult to point out beforehand the different kinds of sensitivity of data.

IdM systems must pay attention to the contextuality of privacy. They need to give individuals the possibility to change privacy settings according to context. This does not simply mean that there is a distinction between 'private' and 'public'. Privacy is not a button which can be switched on, or switched off. Even within the public and private spheres, different privacy perceptions exist. Hence, different privacy features need to make it possible to fine-tune preferences to contextual privacy settings.

Moreover, the individual is not the only actor that determines the privacy-sensitiveness of situations. Social settings have an influence on the use of IdM systems, because understandings of privacy and privacy perceptions vary across social groups, societies, age, and cultures. History and political regime can, for example, influence the perceptions on privacy, just as media coverage, general respect towards government, or recent social debate [Pro06]. In addition, language settings, symbols, and icons are different between societies. IdM systems that need to be adopted broadly, and which want to enhance privacy according to many social settings, need to be adjustable to these settings.

The other way around, the society and legislator can also impose 'norms' on the exercise of privacy that determine occasions in which a claim on complete privacy cannot be accepted. Society and the legislator may therefore impose requirements for accountability to the design of IdM systems. For the adoption of IdM systems, it is important that accountability can be assured in specific instances. Vice versa, society can also not afford that people give up their privacy completely as we have argued in chapter 4. This also means that society has to take the requirements outlined in the Data Protection Directive seriously and not allow people to freely contract away their privacy.

6.2.3.3   Accountability

The first requirement of this chapter, audience segregation, points towards instruments that allow people to create, maintain, and protects partial identities. However, society and legislator may impose restrictions on the identities used by individuals. In some occasions, anonymity, or a specific pseudonym may be undesirable. Hence, just as there are rationales for anonymity, or pseudonymity, there are rationales for identifiability or accountability. One can think here of governmental regulation but also of relationships in which accountability is required, like parent-child relations and employer-employee.

Norms for accountability can be imposed by legal means, but also by social groups. There are thus *de facto* and *de jure* regulatory powers, which may in turn have an extra-territorial effect. Examples here are for instance the regulation considering fraud prevention in multinational organisations, but one can also think of demand for accountability by interest groups, like the public outcry for transparency of the income of CEO's.

Not in all cases it is thus desirable to interact anonymously or with pseudonyms. IdM systems need to take this into account because otherwise they may become considered illegal or undesired. For accountability of an individual, IdM systems must sometimes reveal identities, or credentials must be assigned to ensure that an actor meets to some demands. However, an important condition to implement a mechanism of accountability into a privacy-enhanced IdM system, is that individuals can trust that accountability is only required in concrete and specific occasions.

6.2.3.4 Trust

We have already touched upon the aspect of trust in the requirement of data security. It needs to be stressed that trust and security are not the same. People trust people, not technology [FKH].[5] Therefore, technology can be proven to be trustworthy, but in order for user to actually trust the technology and the relationship with a service provider requires more than just reliable technology. As users will not be skilled to assess the reliability of a technology, this needs to be made transparent and accessible to the user. Furthermore, the look and feel of a technology and markers of quality and functionality are considered important factors in the creation of trust. Especially markers about the authority or credibility of the makers and providers of a service are of importance.

In the online environment, consumers perceive their transactions to be more riskier than transactions in traditional channels. This can be attributed to the fact that the transactions take place without face-to-face contact, but also because much more personal data is disclosed online than offline. Also experience with concrete online transactions is relatively low, the variance in online transaction procedures is much higher than in offline transactions; the steps in a transaction process are often not clear, even though service providers have an obligation to make them clear to the user. It appears that, with a lack of face-to-face contact, users need to rely more heavily on brand name, reputation, past performance, and other information. When designing privacy enhanced identity management solutions it is important to try to understand what the appropriate trust markers are that help people consider the technology trustworthy, provided that the technology is trustworthy of course, and that the data controller can be trusted too (see also [ACC$^+$05]).

6.3 Conclusions

In this chapter we have given a brief high level overview of requirements for privacy-enhancing identity management systems from the perspective of the individual. An extensive and detailed account of these requirement, the legal requirement, and the business requirements can be found in PRIME Deliverable Requirements V3 [KDR$^+$08].

An important aspect of identity management from the perspective of the individual pertains to how individuals present themselves to others. Individuals operate in different spheres and present different aspects of themselves to others in these different spheres because they play different roles and have different interests. The possibility to keep different spheres separated is an important characteristic of modern states. In an online environment this kind of audience segregation requires special attention and implies a number of requirements. A central requirement to facilitate audience segregation is user

---

[5] Even though this may turn out to be a wrong assumption.

control which can be decomposed in a number of more specific requirements. This chapter has briefly elaborated on the various requirements from a joint social/legal perspective, starting with audience segregation to be followed by the ten requirements that together constitute the control requirement: 'comprehension', 'consciousness', 'consent', 'choice', 'confinement', 'consistency', 'context', 'inspection', 'chain control', and 'ex-post user control'. After these, six adoption requirements were discussed: 'social settings flexibility', 'minimize skill level', 'accountability', 'trust in transaction partners', 'trust in communication infrastructure', and 'affordability'.

The requirements discussed are mostly complementary, but on several occasions, a balance between them needs to be struck by the developer and the provider of a service. Privacy, and thus also control and adoption, are dependant on the situation in which a service is implemented.

The requirements presented in this chapter are rather abstract and as such not immediately useful for developers. The PRIME Deliverable Requirements V3 [KDR+08] discusses them in much more detail and also provide measurable targets. For instance, the comprehension requirement (SR.A2 Comprehension) is formulated as: "The user should understand how personal data is handled by the service provider."

Whether the application satisfies this requirement can be examined by answering questions such as:

- Does the application provide sufficiently comprehensive explanations of the consequences of relevant events with respect to the collection and use of (personal) data?
- Does the application provide sufficient general information about (personal) data, its collection and use?
- Does the user understand the application itself?
- Is the user documentation sufficient in scope and understandability?
- Is the user not overloaded with information through too many or too detailed notifications and explanations?

Apart from the legal and user perspective there is also the business perspective to take into account when developing privacy-enhancing identity management applications. The requirements this perspective brings about will be addressed in the next chapter.

7

Privacy-Enhancing Identity Management in Business

Alea Fairchild and Piet Ribbers

Tilburg University

7.1 Introduction

Businesses make use of data routinely for daily operations, including sensitive and/or personal data. Personal data and information are, inter alia, seen as means towards customization of services for employees and for customers.

Some elements of this processing of personal information and some practices have come under increasing scrutiny due to privacy concerns. There is undoubtedly a call for better privacy management in organisations, and a tendency to strengthen privacy regulations and policies up to the point where some of the current processes may even become impossible to execute or become outlawed. However, a basic fact is that even if users want maxmium privacy in business dealings, unless organisations can support these privacy requests, the users will not get their wish.

The PRIME project aims at providing a privacy-enhancing identity management framework that promotes maximum privacy for users within a truly open operating environment. In this respect it operates in an arena that comprises privacy modules of legacy identity management frameworks (HP, IBM, Microsoft), regulatory compliance software (BindView, Computer Associates, NetIQ) and web services-oriented (Higgins, Liberty Alliance) identity management platforms. Because of the stance that PRIME has chosen, its design choices will impact both who and how the PRIME solution will be implemented. For a business or service provider, allowing the user maximum privacy control impacts the services that can be provided. It also impacts the cost and depth of services. The difference between what users want and what enterprises can offer regarding privacy enhanced services implies economic choices of both the user and the firm.

This chapter discusses the business perspective on privacy-enhancing identity management in more detail. It starts with an outline of a business model for privacy enhancement in organisations in section 7.2. This section addresses privacy adoption drivers, a privacy maturity model, risk analysis, and the impact of privacy on business process design. Section 7.3 provides insight in the cost benefit analysis of privacy. Section 7.4 derives a number of business inspired requirements for privacy-enhancing identity management systems. A more extensive version of this chapter can be found in [KDR$^+$08].

7.2   Business Model for Privacy Enhancement

7.2.1   Privacy Adoption Drivers

A central theme in the research on innovation is the way technological innovations are "spread into" a specific environment and how they are subsequently accepted and put to use. This research area is known as 'diffusion and adoption' [Fe05]. Diffusion relates to how innovations are spread across a specific society or industry. Adoption is defined as the process through which a person or organisation evolves from first getting acquainted with the innovation until its eventual full-scale implementation [Re03].

In order to construct a model of PET adoption in organisations we build on Rogers' work on organisational adoption of innovation [Re03]. Rogers distinguishes various variables that influence the process of adoption of innovations. First he describes characteristics of the innovation itself (in brackets their effect on the adoption):

Innovation Characteristics:

- Relative advantage or benefit (+): the advantage offered by the innovation, compared to the practice or technology it is meant to replace
- Compatibility (+): The extent that an innovation resembles its predecessor
- Complexity (-): The effort needed to learn how to use the innovation
- Testability (+): The extent that small scale experiments with the innovation are possible
- Visibility (+): the extent to which the innovation is visible for the outside world

Regarding these items, Rogers notices that their impact is more determined by the subjective perception of these factors by the potential adopter, and not so much by their 'real' value.

Next he distinguishes eight variables that can be considered as characteristic of organisations and their specific openness towards adopting innovation based on Zaltman's work [Ze73].

Organisational Characteristics:

- Top Management's attitude with regard to change: How open is top management to accept the changes that accompany the innovation.
- Centralization: The degree of concentration of power and management
- Internal Organisation complexity: The extent that members of an organisation possess specialized knowledge and expertise.
- Formalization: The level of bureaucracy in an organisation.
- Internal relatedness: The extent that internal member of the organisation are interrelated.
- Organisational slack: The extent that an organisation possesses uncommitted resources.
- Size: The size of the organisation
- Openness: The degree that organisations are in contact with other organisations

Rogers' Diffusion of Innovation [DOI] Theory has gained quite a broad acceptance; the variables have been tested in multiple studies and found to be relevant.

Jeyarai et al. [JRL06] and Fichman [Fic00] found that three clusters of factors explain the organisational adoption behaviour: factors related to the technological innovation, to the adopting organisation, and to the environment of both former factors. They investigated over a hundred variables that have been researched in different studies and performed an empirical test on the best predicting factors for the organisational adoption of IT-based innovations. Combined in clusters, the dominant factors appear top be those related to innovation characteristics, organisational characteristics, and environmental characteristics [TR].

Also other factors appear to influence the adoption process. Fichman [Fic92] argues that adoption of IT based innovations require a different approach than adoption of other technological innovations. Fichman [Fic92], Riverea & Rogers [RR04] and Greenhalgh [Gre04] point to specific effects of innovations in network organisations on inter-organisational relationships.

In our own analysis, we have combined the work of Jearay, Fichman and Rogers in the Conceptual Model, which is shown in Figure 7.1. The distinction of three clusters of factors is based in particular on Fichman [Fic92]. The first cluster (in Fichman's 'terms Technologies & Diffusion environments') relates to the innovation characteristics, including 'Propagating Institutions'. The second cluster (in Fichman's terms 'Technology-Organization combination') describes the relation between the innovation characteristics and characteristics of the adopting organizations. The third cluster (in Fichman's terms 'Organizations & Adoption Environments') encompasses those variables that describe the innovation itself and the specific environment from which it emanates. In the specific case of PETs, this third cluster would encompass in particular privacy policies and regulations, and their level of enforcement

because these factors can be seen as strongly relevant for an adoption decision regarding PETs.
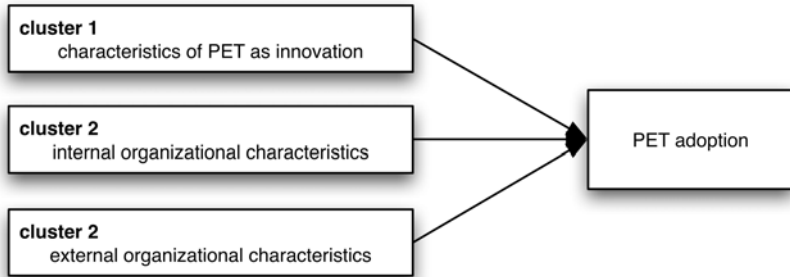
Fig. 7.1 Adoption Model of PET as an innovation

On the basis of the factors discussed above and interviews with a number of experts, we have derived the following factors for the following three clusters.

Cluster 1: Innovation Characteristics

Relative advantage or benefit. The advantage of PET is that it offers a clear privacy protection, which, when properly applied, is compliant with legal requirements. The potential relative benefit compared to other protective measures is large. It, however, appears to be difficult to value these benefits in economic terms, due to the existing ambiguity around PETs and privacy. As a result, these relative benefits may be neglected and enterprises may adopt more conventional measures to accomplish compliance with data protection regulation rather than adopting PETs.

Perceived Complexity. PETs are perceived as complex innovations. The implementation of PETs is thought to require specific expertise in different disciplines. Beyond IT expertise, legal and organisational expertise are required as well; a combination of these competencies is often unavailable in house and may have to be acquired externally.

Perceived Costs. PETs are perceived as expensive innovations (with unclear benefits). Generally PETs will be rightly seen as too complex to apply to be superimposed onto existing systems, with costs perceived as higher than those of traditional measures. If, however, the introduction of PET as an innovation can be envisaged simultaneously with another system innovation, such as when a new system is put into use, then the extra costs of implementing PETs may remain generally at an acceptable level. Thus linking PET introduction with another strong innovation may be the only realistic option.

Role of advisory institutions. Some organisations can play a key role in the diffusion of innovations. The Dutch Data Protection Authority (DPA) has assumed this role with regard to PET in the past, especially with regard to large projects. This role and the 'promotional' attention given to PET by the DPA have had a positive impact on the adoption of this innovation. After this first phase of active support, however, the Dutch DPA stopped promoting the use of PET actively, resulting in lower rate of adoption of this innovation across the country.

Perceived social recognition. The use of PET does not receive a lot social recognition, which is the result of its limited visibility. Also privacy protection is not an issue with which organisations try to differentiate themselves.

Need to integrate PETs into business processes. An important characteristic of privacy enhancing technologies, is that its implementation seems to require an integration in information systems. This requires again legal and technical (IT) expertise. If the PETs indeed needs to be integrated into existing systems, this will lower the willingness to implement them.

Cluster 2: Internal Organisation Characteristics

Complexity of organisational processes. Privacy enhancing technologies usually have to be customized for a specific organisation or process. The more complex this is, the more difficult it is to implement them.

Presence of key persons. The utilization of PETs often depends on specific key persons in an organisation, who are familiar with the concept and take the lead in the adoption process. Such a person has a strong impact on the adoption of PET.

Ties with advisory institutions. The use of Privacy enhancing technologies sometimes depends on the ties that an organisation has with advisory institutions (e.g., DPA). An organisation that has no links with such institutions is not likely to put PETs into use.

Perception of privacy standards. Privacy regulations are often not perceived as being very important and the consequences of non- observance or non-compliance with the law are not always clear nor considered serious. As a result the adoption of PET is not high on the management agenda.

Type of processed data. When the level of legal risk associated with privacy breaches is high a corollary is that there is a bigger incentive to apply PET.

Cluster 3: External Organisational Characteristics (Environment)

Pressure by privacy and data protection laws. Privacy and data protection regulation exert little pressure on organisations to really put PET into use. Only in a few cases does the law specifically refer to PETs,

whereby the decision makers are left free to select alternative protective measures. For instance, the Dutch law imposes quite general and abstract standards. Art. 13 of the Data protection act (Dutch: Wet bescherming persoonsgegevens), for instance only states that 'effective measures' are to be taken, which is subjective or even vague, and provides little direction as to the specific solution to be implemented (also for the IT-auditor). The primary focus of decision makers is on the key business processes. Privacy and data protections often are seen as secondary issues. Generally very little awareness of PETs as privacy tools and practically no demand for privacy audits exists, as no need is felt because there is no felt need.

Complexity of privacy laws. Organisations often do not know/understand what privacy and data protection regulation requires them to do. Because the regulation is seen as overly complex and ambiguous, organisations do not adopt the right set of protective measures to comply with the regulation.

Differences between public and private organizations. In two cases the differences between public and private organizations appear to have been relevant. In the cases of the APK system and the electronic patient file (EPD), this factor has had a negative impact on the adoption of PETs. The reason for this is the considerable initial investment necessary to implement PET. Apparently in the public sector, driven by the interests of the society at large, the justification of this investment is less a problem than in the private sector, which is primarily driven by profit motives. As a result privacy protection is more easily justified in a public sector organization.

Existing offer of PET measures. From our case study analysis this factor came out as having a negative impact on the implementation of PETs. Both the digital customer file (DKD) and the electronic patient file (EPD) were based on standard software offered by known software suppliers like IBM. As a result these organizations depend largely on the functionality available within this software package for their privacy protection and information security. In general privacy protection has not been priority functionality in standard package software. Without adding additional protection the privacy protection will be at the (low) level offered by these packages.

The conceptual model was used in three case studies in the Netherlands.[1] The cases were:

- The digital customer file (DKD), a new development in the Dutch social security system. The DKD should provide citizens, the centres for work and income (CWI), Employers, insurance (UWV) and the local social services access to data about unemployed citizens looking for work.

---

[1] The case studies were carried out by a Master student of Erasmus University as part of a master's thesis. Details about the cases can be found in [WP008].

- The Electronic Patient file (EPD), a new development in the health sector concerning the information services of the health care process. The EPD should provide entities in the health care process access to relevant patient data.
- The APK system. The Dutch state service for traffic (RDW) has to execute the general periodical inspection of vehicles (APK), as required by the regulations on road safety. The purpose of this inspection is to monitor the technical state of vehicles held by Dutch residents. The RWD has outsourced major parts of the APK process to garages meaning that several types of private parties are involved in the APK process.

On the basis of the case studies we arrive at the following augmented adoption model which shows the effect of the various factors on PET adoption.

Table 7.1 The final adoption model: Effect of adoption factors as established in three case studies

Cluster 1 *Characteristics of PETs as innovation*	Effect on adoption
compatibility	negative
complexity	negative
costs	negative
need to integrate PET into business process	negative
Cluster 2 *Internal organisational characteristics*	Effect on adoption
structure and size of the organisation	negative
perception and level of awareness of privacy regulation	positive
diversity in information systems	negative
individual ties with advisory institutes	positive
Cluster 3 *External organisational characteristics*	Effect on adoption
pressure by privacy legislation	positive
differences between public and private organisations	negative
existing offer of PET measures	positive

In summary, we found that only the legal and regulatory pressure (and the promotion by such advisory or supervisory bodies as the data protection agencies (DPA)) with regard to privacy protection is perceived to-date as having an undivided positive impact on the adoption process.

7.2.2   Process Maturity for Privacy

As we have seen, privacy enhancing technologies are not easily adopted by enterprises. The perceived complexity of the implementation as well as the perceived required base line in terms of the perceived required capabilities

of key personnel and management are hindering factors. In other words, the maturity of organisations with respect to privacy and data protection may play a role in the decisions to adopt privacy enhancing technologies. In order to explore this proposition, we investigated how an Identity and Access Management (IAM) maturity model can be adapted to the specific 'privacy adoption maturity' in organisations.

During the last decade, several maturity models have been developed in specific research areas such as business IT alignment, software development and information security. Maturity models describe the maturity of one or more processes within an organization. As a basis for an Identity and Access Management (IAM) maturity model, a number of existing models have contributed to our own model: Nolan Norton's model, the Capability Maturity Model (CMMi), and INK (Instituut Nederlandse Kwaliteit) maturity models. We have extended the IAM model to include a privacy step on top of the existing stages, as companies who are interested in privacy protection have usually already examined identity management issues.

The processes we have defined for IAM are shown in Figure 7.2.

In our IAM model, authentication management and provisioning are mapped on access management since access management deals with authenticating credentials and controlling the access to resources. Given the choice of processes, mainly from work from KPMG [VM01], we have incorporated maturity phases into these processes leading to the IAM maturity model depicted in Figure 7.2.

Identity and Access Management		
Processes		**Technologies**
Authorization Management — Activities aimed at the coupling of users to the already assigned rights to access information and resources through the possible use of authorization models		
User Management — Activities aimed at management of the complete e-identity lifecycle and assigning and revocing of authorizations.	Field of requirements for **Trust** — **Federated Identity Management**	Field of requirements for **Personalization**
Authentication Management — Assigning the correct means of authentication to the user and the management of means of authentication and authentication profiles.		
Provisioning — Propagation of user accounts by means of an automated process or manual process to IT objects.		
Monitoring and Audit — Providing insights into the user accounts, authorizations and process execution. Achieved by using logging, permanent auditing and reporting.		

Fig. 7.2 Identified Identity and access management processes and technologies

Table 7.2 Identified Identity and access management processes and technologies

	Immature	Starting-up	Active	Pro-active	Top class
Authentication management	no authentication means	authentication formulated arbitrarily (authentication means are provided, adjusted and deleted on user request	Authentication requirements based on a one time survey	Authentication requirements based on continuous risk analysis	Authentication requirements based on continuous risk analysis and are continuously adjusted
User management	Double and inconsistent entries because of ad hoc processes	Entries can be double but they are consistent	Central registration, limited user group, manual procedures	Central registration, controlled authorization processes, manual procedures	Central real-time controlled authorization sources, automated procedures
Authorisation management	No authorisation matrices, authorization is defined ad hoc	Authorization matrices defined but not updated	Authorisation matrices are updated periodically	Role based access control used for control for critical applications	Role based access control for ll applications and continuous updated authorizations
Provisioning	Manual process locally	Limited automated unreliable processes locally	Limited automated but reliable processes locally	Limited automated and reliable and multiple sources	Automated and reliable for multiple sources
Monitoring (audit)	No responsibility delegated into a AO/IC organization	Sporadically delegated responsibility of AO/IC	Partial delegation of responsibility to AO/IC	Full responsibility to to AO/IC	Full responsibility to A/IC with periodic reporting

Based on the phase characteristics depicted in Figure 7.2 and the description of the phases provided by the different models, the following general phase descriptions are induced:

Phase 1: Immature. Only a few processes have been defined and processes are conducted on an ad hoc base. In this phase the notion of 'Identity and access management' begins to enter within the organisation. No or very little applications or processes are in place to facilitate IAM. Monitoring and audit are virtually nonexistent and provisioning is performed manually. Means of authentication are very rudimentary and limited to, e.g., local username and/or passwords. User profiles are maintained locally and can be duplicated and inconsistent. Authorizations are not regulated, are assigned only upon adhoc request and are not based on an authorization matrix. This leads to a situation in which different user profiles could be in place, e.g., in the company's database and on the individual computer, where it serves to provide access to the complete array of programs installed on that one pc. Provisioning is done manually at each workstation as a central solution is most likely not available at this stage. User profiles are only updated locally by the administrative personnel and the profiles on the workstations are either maintained by the employee themselves or not at all.

Phase 2: Starting up. Processes that seem to work and be in order, are repeated. In the second stage of maturity the company is starting to realize that IAM is needed. Authorization matrixes are developed and authentication requirements are arbitrarily formulated based on user requirements. identity databases are improved to the point, that they may still contain double entries, but without any inconsistencies. Provisioning activities are becoming automated but are still done locally. Monitoring and audit activities are getting started although in a highly sporadic fashion and responsibility is only sometimes delegated to AO/IC. At that stage, however, these new activities such as automated distributed provisioning and the creation of authorization matrixes and authentication requirements are not yet very reliable nor periodically updated.

Phase 3: Active. Processes are standardized and documented to review if they are executed accordingly. Maturity phase three is in essence an improvement on phase two. Most of the processes are still the same, but are executed regularly or have become regulated. Authentication management has improved significantly since it is no longer based on ad hoc user requirements, but on a one-time survey. User management also has further improved: Users are now registered centrally and positioned user groups. Provisioning is still limited to a certain number of applications and executed locally, the automated provisioning however has become more reliable. The responsibility of the IAM processes is increasingly delegated to the Monitoring and Audit activities.

Phase 4: Pro-active. Performance and success are measured and quality measures are done. In this phase the authentication requirements are updated periodically based on risk analyses regularly performed. While user management is still a manual process it is now a totally centralized process controlling all user registrations. Authorization management is characterized by the introduction of techniques such as role based access control (RBAC) for critical applications. This means that the access rights assigned to the user are based on the access rights given to the group. Provisioning is not only automated and reliable but the scope of provisioning is enlarged from local to multiple provisioning sources. Responsibility for Monitoring and audit becomes the total responsibility of the AO/IC organisation.

Phase 5: Top class. Processes are systematically improved with the help of quantitative feedback of results, test results and innovative ideas. The general improvement for this maturity level entails continuous improvement and/or adjustment of the IAM processes. The great change for user management is that authorization processes no longer have to be done manually, but now become automated. Authorization management is changed in the way that RBAC is now implemented for all application and authorization rules are adjusted real-time. Provisioning has become automated and reliable for all provisioning sources. Monitorcontrols now but also acts on its control activities by regular reporting.

7.2.2.1   Incorporating Privacy in the Maturity Model

In the Whitebook on Privacy Enhancing Technologies [KGH$^+$04], privacy enhancing technologies are presented as a compound of several technologies which can be divided in four different stages (shown in Figure 7.3). Obviously these technologies require a certain level of IT infrastructure. PETs also requires a solid foundation in the form of Identity and access management, so as to minimize the use of and access to sensitive personal data. This is clearly reflected by the mention of the technology 'Secured Access' among the general PET measures in Figure 7.3. Secured Access, however is only a first step to achieve proper privacy enhanced systems. Privacy Enhancing Technologies also strive among others, to ensure right protection of a person's identity by the segregation of sensitive information and also by such measures as immediate information removal after use (or even not capturing the information in the first place.), whenever the information needs of the organisation will make such measures realistically possible.

In order to implement privacy enhancing technologies, a certain maturity of the organization is required. It is highly unlikely that immature organisations have a strong awareness of privacy protection, let alone will implement significant privacy enhancing technologies. The IAM maturity level may therefore provide a strong indication for the readiness of an organization regarding PET implementation.
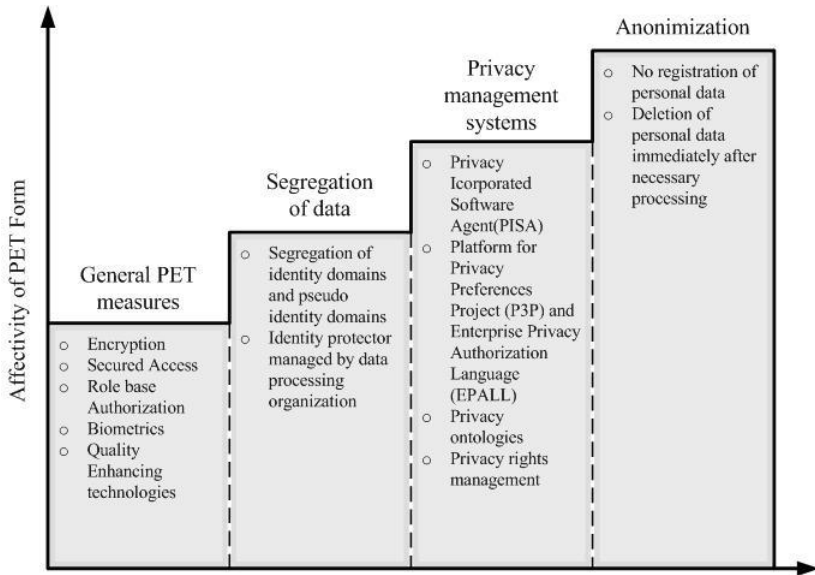
Fig. 7.3 Staged effectivity of PET including used technologies per stage

Next to user management, authentication management and authorization management, provisioning and monitoring and audit can also play an important part in a PET implementation. For instance, when a central database of information is accessed by different organisations, provisioning (automated or not) can play an important to keep user accounts for that database up to date at the different locations. Monitoring and Audit plays an important role when reviewing the current status of user accounts and controlling if data is accessed by authorized users only. Thus depending on the requirements of the organization on its PET implementation, a certain level of maturity is needed for the relevant IAM processes. By combining the PET stages and the maturity model, the maturity model can predict when PET will be appropriate in which stage of organizational development.

On the basis of the model discussed so far, we may predict that PETs are more likely to be implemented by organisations in the Top Class and Pro-Active maturity levels than in more immature organisations, with the exception for organisations that update authorization matrices periodically (organization at the level: active). There are exemptions for those organization that belong to the category of (micro/mini) SMEs where trust is a critical success factor, like in the medical profession, barristers, notaries etc. Although the processes mentioned in the maturity model are likely to be non-existent in these situations, it may be expected that those SMEs will protect personal information of their clients encrypted or will use rudimentary PET tools.

7.2.3   Risk Analysis for Data Privacy

Privacy management in organisations requires procedures to protect against unauthorised access and usage of personal data. To determine the appropriate level of security an organization has to implement, the state of the art and the costs of implementation, as well as risks and effects associated with the processing, and the nature of the data to be protected have to be taken into account.

The level of security that a controller must provide will depend on the risk class. Article 17 of the Data Protection Directive (95/46/EC), article 4 of the ePrivacy Directive (2002/58/EC) and the Communication from the Commission to the European Parliament and The Council (COM (2007) 228 final form the basis for the use of Privacy-Enhancing Technologies. PETs consist of ICT measures protecting informational privacy by eliminating or minimizing personal data or by preventing the unnecessary or undesirable processing of such data, without compromising the functionality of the information system. They are more than just appropriate technical measures; they are means to systematically ensure compliance with the Privacy Directives.

In the protection of personal data form a security standpoint, it is important that the measures taken address realistic threat given the nature of the data concerned and the scale of the processing activities. The risk may be regarded as the product of the likelihood of an undesirable event and the seriousness of the implications of that event. The greater the risk, the stricter the protection requirements that must be met. As a guide to the measures that are appropriate, data processing procedures are divided into a number of predefined risk classes. Each class is linked to a particular level of protection. The main factors influencing the level of protection required include:

- *The significance attached by society to the personal data to be processed.* Specific combinations of personal data, size and objective of processing and types of utilization may result in an increased level of sensitivity. The privacy directive 95/46/EC qualifies these data as special and sensitive personal data, which justify an increased level of protection and a different way of processing (e.g., PET). Sensitivity can increase because of multiple reasons. First, because of the potential consequences for those, whose data have been used or processed in a careless or unauthorized way. Second, sensitivity (and risk) can increase when the amount of data and the complexity of processing increase. The more data that are contained in a database, and the more complex the processing (about different persons, profiles built up during longer period of times), the higher the level of available information, and so also the higher the probability of inaccurate or unauthorized use. Third, the type of use plays a role. Especially relevant are the frequency of consultation (once a year versus multiple times a day) and the number of locations from which access is possible.
- *The level of awareness within the processing organization regarding information security and the protection of personal data and subjects' privacy.*

This factor relates in particular to the level of (privacy) maturity in an organization. To what extent are people aware of privacy risks and act accordingly?

- *The nature of the ICT infrastructure within which the personal data is to be processed.* The ICT infrastructure will differ per organization in terms of being state of the art, complexity, technical possibilities and types of use. The following factors have to be taken into account:
  - Characteristics and organization location of IT equipment;
  - Types of computer networks in use;
  - Databases and retrieval systems in use for personal data;
  - Architecture of information systems and processes in use for personal data.

The controller must perform a thorough analysis. A Privacy Impact Analysis (PIA) or Privacy Threat Analysis forms the basis for assessing the types and levels of risk associated with the processing of specific categories of personal data. On the basis of these findings, the controller can decide which risk class the intended procedure falls into and what level of protection is therefore required. The analysis must be verifiable and it must be possible to give an account of the analysis if necessary. Four risk classes are recognized [BB01, p.21]:

Risk class 0: Public-level risk. The personal data to be processed is already in the public domain. It is generally accepted that use of the data for the intended purpose represents no risk to the subjects. This document therefore proposes no special protection measures.

Risk class I: Basic-level risk. The consequences for the subjects of the loss or unauthorized or inappropriate use of their personal data are such that standard (information) protection measures are sufficient.

Risk class II: Increased risk. The loss or unauthorized or inappropriate use of the personal data would have additional consequences for the subjects. For instance, certain types of personal data referred to in Article 8 of the Data Protection Directive 95/46/EC enjoy special legal protection and therefore require at least the level of protection associated with this risk class. The types of personal data in question are data concerning a data subject's religion or philosophical beliefs, race, political opinions, health, sex life, trade-union membership, criminal record or record of unlawful or antisocial behavior following the imposition of an injunction.

Risk class III: High risk. Where several collections of special categories of personal data are to be processed, the potential consequences of processing can be sufficiently serious for the data subjects that the procedure warrants inclusion in risk class III. The measures taken to protect data processed in a class III procedure must meet the highest standards.

The interrelationships between the various risk classes can be summarized as shown in Table 7.3.

Table 7.3 Risk levels in privacy [BB01]

		Nature of personal data		
Quantity of personal data	*Nature of the processing*	Personal data	Sensitive personal data (in accordance to 7-8 EU 95/46/EC)	Personal data of financial and/or economic nature
Small	Simple	Risk class 0	Risk class II	Risk class II
Large	Complex	Risk class I	Risk class III	

7.2.4   Privacy Impact on Business Process Design

There is a relationship between the need for privacy and the level of 'person-related information' intensity in an organizational process, including organizational maturity to handle privacy along with the associated risk levels of the process. The more 'person-related information' is handled within an organization, the higher the possibilities of risk associated with privacy and information loss or modification. The ability to adequately handle privacy risk depends on the maturity of the organization and how capable the organization is in organizing both technological and organizational measures to ensure privacy in the process.

The starting point for designing privacy aware business processes, is to develop a privacy policy derived from the organisation's objectives that can serve as the basis for policies regarding the processing of personal data. The processing policy has to include elements such as corporate methods of utilizing passwords or other identity mechanisms, and must give tangible form to specific measures and procedures for the processing cycle of personal data. Defining tangible measures and procedures occurs after thorough risk analysis, and a complete 'inventory-listing' of the threats which processing of personal data is exposed to as discussed in the previous section. Within this context the strong and the weak points of data processing are laid down. The risks together with the strong and the weak points of the processing organisation and a cost-benefit analysis, based on the defined privacy policy, result in a carefully considered choice for the organisational and technical measures to be taken.

Management must, with the help of a monitoring system, examine to what extent the measures taken fulfil the objectives of the formulated privacy policy.

Management must indicate in what way and with which intensity it wishes to receive the monitoring data. The results of the performed monitoring form the basis for any corrective actions, adjustment of measures and procedures taken or adjustment of the formulated policy.

An organisation's management can determine the way in which technical and organisational measures are taken in order to safeguard the protection of personal data. It will try to adapt this to the existing organisation and further detailing of administrative organisational and technical measures and procedures to safeguard (automated) data processing. Based on the existing set of control instruments, the management can further implement the legal requirements in an effective and efficient way. The law currently does not impose organisations any compulsory set up with regard to these technical and organisational measures. However, an organisation can organise a Privacy Audit to check how well the organisation addresses what regulatory issues there are in their industry. A Privacy Impact Assessment (PIA) examines the privacy issues of a project or policy and helps to manage privacy impacts from a privacy perspective. It is an assessment tool which shows the flows of personal information.

In the framework of the Privacy Audit, the following quality aspects are relevant for the compliance with the requirements arising from the compliance monitoring:

Exclusivity / Confidentiality: Only authorised people have access to and can make use of personal data.
Integrity: The personal data must be in accordance with the projected part of reality and nothing may be wrongfully held back or made to disappear.
Continuity: The personal data and the information derived from this must be available without restrictions in accordance with the agreements made to that respect and the existing legal regulations. Continuity is defined as 'undisturbed progress of data processing'.
Auditability: Auditability is the extent to which it is possible to gain insight into the structure (documentation) and operation of an object. The quality aspect audit ability also encompasses the extent to which it is possible to determine that processing personal data has been carried out in accordance with the requirements with regard to the aforementioned quality aspects.

The extent to which these aspects must be used in a concrete situation partly depends on the risk analysis performed by the auditor. The choice for quality requirements per audit object must be explained in the audit plan. The extent to which the quality aspects mentioned are relevant for obtaining a certificate will be worked out in the certification scheme.

The process of protecting personal data starts with the completion of a Privacy Impact Assessment (PIA)[2], or Privacy Threat Analysis or the use of the Privacy Diagnostic Tool (PDT)[3]. A PIA seeks to set forth the essential components of any personal information system. The PIA and PDT will identify the threats and risks that will define and identify the solution parameters. The privacy threat analysis (that have been developed in the EC funded PISA project) has modified the risk assessment method for information security formulated in British Standards 7799 (ISO 17999) (The Code of Practice for the Risk Analysis and Management Method) to the needs of the protection of personal data. Also the following approaches for a privacy risk assessment can be used in a modified way (taken into consideration the specific requirements of the privacy legislation): the risk assessment of the Information Security Handbook of the Central Computers and Telecommunications Agency (CCTA) or Information Technology Security Evaluation Criteria (IT-SEC), as published by the European Communities in 1991. Without a privacy threat analysis/PIA it is impossible to implement the appropriate technical and organizational measures to prevent privacy intrusions, the loss of personal data and any form of unlawful processing.

7.3 Cost Benefit Analysis of Privacy

Privacy protection is currently seen as a negative cost driver in a cost/benefit analysis. In fact it represents a driver that is a bit neutral in that it may both cause some costs and avoid other costs. Good privacy protection, whenever properly communicated, will generally establish trust, which is a basic driver for a 'better image' with customers and business partners and consequently for improved revenues [PJBR06]. It is clear that a sound business case for PETs should investigate their implications both on costs and on revenues.

Investments are long term commitments of resources made in expectation of future revenues. Costs and benefits associated with investments can either be tangible - which means they can be assessed and a value expressed for instance in monetary terms - or intangible [Pis01], which means those non-monetary elements that cannot be seen, touched or physically measured.

In principle tangible costs and benefits are easy to calculate; examples are savings of labor and other costs, productivity improvements, and revenues. Intangible costs and benefits, which are not directly expressible in monetary terms, are more difficult to quantify. Examples of intangible elements are brand advantage, which reflects the change in brand awareness and reputation to be expected from the investment; or the competitive advantage, rsulting

---

[2] See D.H, Flaherty, Privacy Impact Assessments: An Essential Tool for Data Protection, in One World, One Privacy, Roma (Garante per la Protezione dei Dati Personali) 2000.

[3] See the Privacy Diagnostic Tool Workbook version 1.0 developed by the Office of the Information Commissioner of Ontario: www.ipc.on.ca/.

from an ability to respond more effectively to competition. Although intangibles are not directly expressible in monetary terms, with the use of pseudo calculations their perceived impact on the organization's (strategic) objectives can be expressed with a number on some scale. In principle in an investment analysis tangible costs and benefits should be analyzed as far as possible. At the point where this analysis stops, because of lack of data, intangible impacts should be identified and categorized.

Questions that should be raised in order to perform a cost/benefit analysis start with [KvGtH+04]:

- Do PETs make an essential contribution to the policy targets and objectives of the organization?
- What tangible and intangible benefits can PETs achieve in the organization?
- What are required investments and structural costs for PETs?

Drilling down these questions to more concrete questions, we arrive at questions such as [SAS06]:

- How much is the lack of privacy costing the business?
- What impact do privacy breaches have; what would the damage be of a catastrophic privacy breach?
- How much do privacy protective measures cost?
- What are cost-effective solutions?
- What are the potential benefits of PETs and can they be quantified?
- How does privacy protection contribute to the competitive position of the firm?

We have adapted several models of investment analysis to create a model called ROPI - return on privacy investment, which adds several variables to the calculation for security investment, based on privacy breach estimates. We propose the following cash-flow components:

Annual Loss Exposure (ALE) [SAS06] is the multiplied projected costs of a privacy incident and its annual rate of occurrence. Basically this encompasses revenue losses, legal claims, productivity losses because of privacy breaches, repair costs and lost business.

Reputation Recoverage Costs (RRC) contain those expenses needed to restore the reputation of the company damaged by privacy breaches; examples are additional costs for PR and Marketing. Moreover if a privacy breaches affects the share price of the company (see ChoicePoint, Double Click), possibly breaches affects the share price of the company (see ChoicePoint, Double Click), possibly additional financial guarantees may be required by banks and other financial institutions.

Expected Revenue Accrual (ERA) represents, on the positive side, possible marketing impacts on market share and revenue of publicized implementation of PETs.

Recurring Privacy Costs (RPC) contains the yearly (additional) privacy
costs caused by the proposal; this will encompass needed PIAs, audits,
privacy officers etc.

The cost/benefit analysis has to compare the situation with the PET(s)
in place, with the situation without PET(s) in place. Basically this comes
down to analyze the cash-flow differences between the two situations. This
can be done either by applying a factor RM (Risk Mitigated) to the situation
without the investment or by subtracting the full expected cash flow of the
two situations from one another. The resulting NPV (Net Present Value) of
a privacy protection solution is consequently as follows:

$$NPV = -I(p) + \sum_{j=1}^{n} \left\{ (ALE + RRC) \cdot RM + ERA - RPC \right\} / (1+i)^j \ ,$$

where
NPV = Net Present Value,
n = maximum lifespan of the project in number of periods (usually a year),
j = represents a period (a year),
i = represents the minimum required return, and
$\sum$ = represents summation of terms during the indicated period (j = $1, \ldots, n$).

The NPV is the difference between the cash proceeds and cash-outlays
discounted at the minimum required return (i) during the lifespan n (j =
$1, \ldots, n$, where j represents a period, usually a year) of the investment project.
From the NPV perspective an investment will be acceptable if the NPV is
equal to or greater than 0, which means that return on the cash outlays meets
the required minimum. Consequently, a NPV smaller than 0 leads to rejection
of the project.

The discounted cash-flow procedures, like the NPV, measure cash flows
in terms of a required rate of return (hurdle rate, cost of capital) i to deter-
mine the acceptability of the investment project. The cost of capital refers
to the rates of return expected by those parties contributing to the financial
structure: creditors and shareholders. It represents the costs of funds used to
acquire the total assets of the firm. It is generally calculated as a weighted
average of the costs associated with each type of capital (long term loans,
short term loans, and equity) included in the financial structure of the firm.

Based on this formula we may conclude that investments in PETs should
be justified by:

- reduced annual loss expectancy,
- reduced reputation recoverage expenses,
- expected revenue accruals (due to 'PETs Inside'), and
- reduced recurring privacy protection costs.

The above approach can be applied in various ways:

- A straightforward calculation of the NPV of an anticipated project; if the NPV is greater than or equal to zero the project is acceptable.
- Starting from the right term in formula with estimates the yearly cash flow a maximum for the investment I(p) can be derived.
- Consequences of possible events, like a serious privacy breach, can be estimated and allow to give an estimate of a maximum necessary investment.

Given the uncertainties about exact data in the model, calculating the outcome under different assumptions (scenarios) gives a better insight in the behaviour of cost and revenue figures and their impact on the eventual outcome. Analyzing different scenarios provides managers with a systematic approach for decision making about the application of PETs.[4]

7.4 Requirements from a Business Perspective

In this section we discuss requirements for the development of Privacy-enhancing Identity management from the perspective of the economic implications for service providers. This includes the impact of maximum privacy controlled by the user on economic choices for the service provider, mainly in the context of the business transaction and associated processes. The Requirements focus on the value of a Privacy-enhancing Identity management framework on the perceived and realised value of privacy enhancement to business processes, in the form of the cost, quality and process impact of the privacy-enhancing features. This will include business infrastructural issues, on what resources of the firm might be needed to implement and maintain the PET framework. This includes:

Technology aspects: Financial and technological impact of the inclusion of the privacy solution to the greater identity management architecture. This is particularly important in ongoing production environments with legacy applications.

Data: Economies of scale on data management, based on how the module is deployed, how application dependent it is, how reusable the data context is.

In our assessment of both the Integrated and Application prototypes of PRIME, we took the view that the enterprise adopting PET would want to add a privacy module within an IDM framework or solution. To be accepted in the organization, this Privacy-enhanced IDM should build upon and increase the value of existing data processes, as well as promote data quality improvements in line with international standards and good practice. We examined quality management issues of privacy in terms of Boehm's seven quality factors [BBK+78] [5] that together represent the qualities expected from a software

---

[4] Examples of privacy cost/benefit analysis can be found in [WP008].

[5] Portability, reliability, efficiency, usability, testability, understandability, flexibility.

system. We also looked at the necessary changes required to implement privacy as possible economic impacts to the data quality of the business process. This resulted in the following infrastructural requirements (IRs):

Cost to implement: An analysis, based on business infrastructural issues, on what it might cost to implement. This includes technology cost, data cost, and indirect costs, such as education and training. If the PIM is, like PRIME, a middleware solution, one cost element is where in the organization it will be deployed.

IR-1: *The cost to implement should support the business case and should not be prohibitive.*

Cost to integrate: This is primarily an assessment of process change costs, both to the application and the business. Examples of the types of activities that users must perform when changing processes will include matching application business logic to the enterprise processes, creating new processes and the resulting configurations based on enterprise needs. In general integrating PET into legacy systems is problematic.

IR-2: *The architecture should enable a smooth integration into existing business processes and systems.*

Impact on current process efficiency: Organisations would perform a limited process audit. This examines what changes would be necessary to the processes and what necessary resources would be needed to create performance of process to necessary level, either technologically or via organizational changes.

IR-3: *The application of PRIME should not impede current process efficiency.*

Transparency: Level of awareness for both the user and the organization of the organization's privacy policy. In the user context, how visible is the controlled authentication access to only what is needed for the business transaction. Also the ability to track different versions of privacy policy as changes are made. This means an examination of the firm's privacy policy and the economic impact of the transparency of such a policy.

IR-4: *How PRIME is linked to the corporate privacy policy should be visible and auditable.*

Scalability: For the organization, this relates to the financial and process impact of scaling the privacy solution on a wider scale than the initial implementation / application. Also the possibility of an initial deployment for one application into a larger implementation requires the appropriate economies of scale.

IR-5: *PRIME should be easily scalable from a local to an enterprise wide implementation.*

Modularity: For the organization, where is the module deployed, and how application independent is the implementation for business purposes. This includes simple, consistent experience while enabling separation of contexts through multiple operators and technologies.

IR-6: *The PRIME architecture should be modular.*

Architectural fit: Financial and technological impact of the inclusion of the privacy solution to the greater identity management architecture. This is important if the user already has a product suite investment, and is considering the privacy module as an add-in.

IR-7: *PRIME should be compatible with existing Identity Management Architectures.*

Environmental fit, data types: How standard the privacy solution is, does it fit with the technological choices already made by the firm, ability for plug-ins, etc.

IR- 8: *PRIME should easily fit with existing system and data architectures.*

7.5 Conclusion

This chapter focused on privacy-enaching identity management from a business perspective. Where the law is basically an external motivator for privacy protection, we looked at privacy protection from an internal company perspective. The basic question we explored concerns economic motivations for an organization to invest in privacy and identity management. For this reason we started with an analysis of technology adoption processes. This analysis showed that there are few intrinsic drivers for enterprises to adopt privacy-enhancing technologies. Next we looked at a maturity model regarding identity management processes. This model suggests that we may only expect enterprises that are sufficiently mature regarding their identity management and that are sufficiently privacy aware to be able and willing to implement advanced privacy-enhancing technologies.

An enterprise, apart from being capable of implementing PETs, also has to make a cost/benefit analysis in order to decide to invest in PET implementation. Privacy protection is currently seen as a negative cost driver in a cost/benefit analysis. This may be too myopic. Privacy protection may cause some costs and avoid other costs. Furthermore, an analysis of the risk level of data breaches or data loss, may provide causes for re-assessing the costs and benefits. Good privacy protection will also, when properly communicated, generally establish trust, which is a basic driver for a 'better image' with customers and business partners and consequently for improved revenues [PJBR06]. It is clear that a sound business case for PETs should investigate their implications both on costs and on revenues. This chapter has provided a model for determining the costs/benefits of investing in privacy-enhancing technologies.

Finally, it has provided a set of eight requirements for PRIME as an infrastructural investment (so called 'IRs'). These requirements are related to the economic deployability of PRIME as software in an infrastructure.

References

[ACC⁺05] Andersson, C., Camenisch, J., Crane, S., Fischer-Hübner, S., Leenes, R., Pearson, S., Pettersson, J.S., Sommer, D.: Trust in Prime, pp. 552–559 (2005)

[AG05] Aquisti, A., Grossklags, J.: Privacy and rationality in individual decision making. IEEE Security & Privacy, 24–30 (January/February 2005)

[Agr97] Agre, P.E.: Introduction. In: Agre, P.E., Rotenberg, M. (eds.) Technology and Privacy: The New Landscape. The MIT Press, London (1997)

[Ake70] Akerlof, G.: The market for lemons: Quality uncertainty and the market mechanism. Quarterly Journal of Economics (84), 488–500 (1970)

[Alt75] Altman, I.: The Environment and Social Behaviour. Brooks/Cole, Monterey (1975)

[And05] Anderson, J.Q.: Imagining the Internet: Personalities, Predictions, perspectives, lanham. Rowman & Littlefield (2005)

[APW07] APWG. Phishing activity trends: Report for the month of december. Technical report (2007), www.antiphishing.org

[BB01] van Blarkom, G.W., Borking, J.J.: Beveiliging van Persoonsgegevens (av 23), Den Haag (2001)

[BBK⁺78] Boehm, B., Brown, J.R., Kaspar, H., Lipow, M., McLeod, G., Merritt, M.: Characteristics of Software Quality. North-Holland, Amsterdam (1978)

[BGS05] Berendt, B., Gunther, O., Spiekermann, S.: Privacy in e-commerce: Stated preferences vs. actual behaviour. Communications of the ACM 48(4), 101–106 (2005)

[BJ02] Blanchette, J.-F., Johnson, D.G.: Data retention and the panoptic society: The social benefits of forgetfulness. The Information Society 18, 33–45 (2002)

[Blu02] Blume, P.: Protection of Informational Privacy. Copenhagen (2002)

[BMK07] Brody, R.G., Mulig, E., Kimball, V.: Phishing, pharming and identity theft. Academy of Accounting and Financial Studies Journal 11(3) (2007)

[Byg01] Bygrave, L.: Core principles of data protection. In: PLPR, 9 (2001)

[Byg02]   Bygrave, L.: Data Protection Law — Approaching its Rationale, Logic and Limits. Kluwer International, Dordrecht (2002)

[Cam05]   Cameron, K.: The laws of identity. Technical report, Microsoft Corporation (2005)

[Car02]   Carey, P.: E-Privacy and Online Data Protection, Butterworths (2002)

[Cav06]   Cavoukian, A.: 7 laws of identity, the case for privacy-embedded laws of identity in the digital age (2006)

[Cha06]   Chappell, D.: Introducing windows cardspace (2006)

[Cho06]   Choo, K.-K.R.: Issue report on business adoption of microsoft passport. Information Management & Computer Security 14(3) (2006)

[Cla94]   Clarke, R.: The digital persona and its application to data surveillance. The Information Society 10, 77–92 (1994)

[CMBG$^+$02]   Mont, M.C., Bramhall, P., Gittler, M., Pato, J., Rees, O.: Identity management: a key e-business enabler (2002)

[Coy07]   Coyle, K.: Identity crisis. The Journal of Academic Librarianship 33(4), 512–514 (2007)

[DB04]   Donath, J., Boyd, D.: Public displays of connection. BT Technology Journal 22(4), 71–82 (2004)

[DD08]   Dhamija, R., Dusseault, L.: The seven flaws of identity management: Usability and security challenges. IEEE Security and Privacy 6(2), 24–29 (2008)

[DG02]   Danna, A., Gandy, O.H.: All that glitters is not gold: Digging beneath the surface of data mining. Journal of Business Ethics 40, 373–386 (2002)

[DG04]   Dumortier, J., Goemans, C.: Privacy protection and identity management. In: Jerman-Blazic, B., Schneider, W., Klobucar, T. (eds.) Security and Privacy in Advanced Networking Technologies. IOS Press, Amsterdam (2004)

[DGZP05]   Dutton, W., Guerra, G.A., Zizzo, D.J., Peltu, M.: The cyber trust tension in e-government: Balancing identity, privacy, security. In: Public Administration in the Information Society: Essays on Risk and Trust, vol. 10 (2005)

[DH01]   DiMaggio, P., Hargittai, E.: From the 'digital divide' to 'digital inequality': Studying internet use as penetration increases, p. 19 (2001)

[DL01]   van Duivenboden, H., Lips, M. (eds.): Klantgericht werken in de publieke sector: introductie, Lemma, Utrecht (2001)

[Don98]   Donath, J.S.: Identity and deception in the virtual community. In: Kollock, P., Smith, M. (eds.) Communities in Cyberspace. Routledge, London (1998)

[DS97]   Dammann, U., Simitis, S.: EG-Datenschutzrichtlinie. Nomos Verlagsgesellschaft (1997)

[EDP07]   EDPS. Opinion on the communication from the commission to the european parliament and the council on the follow-up of the work programme for better implementation of the data protection directive. Official Journal of the European Union, C255 (2007)

[EDP08]   EDPS. Opinion on the proposal for a directive of the european parliament and of the council amending, among others, directive 2002/58/ec concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications) (2008)

[EI06]   EPIC and Privacy International. Privacy & Human Rights, Washington (2006)

[EIA07]   EIAA. Mediascope europe study 2007 (2007)

[Etz99]   Etzioni, A.: The Limits of Privacy. Basic Books, New York (1999)

[Eur05]   Eurobarometer. Social values, science and technology. Technical Report 225, European Commission (2005)

[Fe05]   Fagerberg, J. (ed.): The Oxford Handbook of Innovation. Oxford University Press, New York (2005)

[Fic92]   Fichman, R.G.: Information technology diffusion: A review of empirical research. In: ICIS, pp. 195–206 (1992)

[Fic00]   Fichman, R.G.: The Diffusion and Assimilation of Information Technology Innovations. In: Framing the Domains of IT Management: Projecting the Future ... Through the Past, Pinnaflex Educational Resources, Inc. (2000)

[FKH]   Friedman, B., Kahn, P., Howe, D.: "trust online". Communications of the ACM 43, 34–40

[Fou77]   Foucault, M.: Discipline and Punish: The Birth of The Prison. Allen Lane, London (1977)

[Fri68]   Fried, C.: Privacy. The Yale Law Journal 77, 475–493 (1968)

[Gan93]   Gandy, O.H.: The Panoptic Sort. A Political Economy of Personal Information. In: Critical Studies in Communication and in the Cultural Industries. Westview Press, Boulder (1993)

[Gav80]   Gavison, R.: Privacy and the limits of law. The Yale Law Journal 89(3), 421–471 (1980)

[GKM07]   Gilbert, D., Kerr, I.R., McGill, J.: The medium and the message: personal privacy and the forced marriage of police and telecommunications providers. Criminal Law Quarterly 51, 469–507 (2007)

[Gof59]   Goffman, E.: The Presentation of Self in Everyday Life. Doubleday Anchor Books, Garden City (1959)

[Gre04]   Greenhalgh, T.: Diffusion of innovations in service organisations: Systematic review and recommendations. The Milbank Quarterly 4, 581–629 (2004)

[Gri08]   Grimmelmann, J.: Accidental privacy spills. Journal of Internet Law (2008)

[Han08]   Hansen, M.: User-controlled identity management: the key to the future of privacy? Int. J. Intellectual Property Management 2, 325–344 (2008)

[Hen08]   Heng, J.: Idc: 45gb of data held on every person. CIO (2008)

[Hey02]   Heylighen, F.: Complexity and information overload in society: why increasing efficiency leads to decreasing control (2002)

[Hil06]   Hildebrandt, M.: Privacy and identity. In: Claes, E., Duff, A., Gutwirth, S. (eds.) Privacy and the Criminal Law, Intersentia, Antwerpen (2006)

[HS03]   Holznagel, B., Sonntag, M.: A case study: The janus project. In: Nicoll, C., Prins, J.E.J., Van Dellen, M.J.M. (eds.) Digital Anonymity and the Law — Tensions and Dimensions. TMC Asser Press, The Hague (2003)

[Int97]   Introna, L.D.: Privacy and the computer: why we need privacy in the information society. Metaphilosophy 28(3), 259–275 (1997)

[JB05]   Jutla, D.L., Bodorik, P.: Sociotechnical architecture for online privacy. IEEE Security & Privacy, 29–39 (2005)

[JRL06]   Jeyaraj, A., Rottman, J.W., Lacity, M.C.: A review of the predictors, linkages, and biases in it innovation adoption research. Journal of Information Technology 21(1), 1–23 (2006)

[KDR$^+$08] Kosta, E., Dumortier, J., Ribbers, P., Fairchild, A., Tseng, J., Liesebach, K., Franz, E., Leenes, R., Hoogwout, M., Priem, B., Kölsch, T., Zibuschka, J., Kramer, G., Schumacher, G.: Requirements for privacy enhancing tools (May 2008 version). Technical report, PRIME Consortium (2008), https://www.prime-project.eu/

[KGH$^+$04] Kroon, R., van Gils, H., ter Hart, J., Overbeek, P., Tellegen, P., Borking, J.: Privacy enhancing technologies – witboek voor beslissers. Technical report, Ministry of internal affairs and Kingdom relations (2004)

[KL05]   Koops, B.-J., Leenes, R.: 'code' and the slow erosion of privacy. Michigan Telecommunications and Technology Law Review 12(115), 115–188 (2005)

[KL06]   Koops, B.-J., Leenes, R.: Identity theft, identity fraud and/or identity-related crime. Datenschutz und Datensicherheit 30(9), 553–559 (2006)

[KLM$^+$08] Koops, B.J., Leenes, R., Meints, M., Van der Meulen, N., Jacquet-Chiffelle, D.-O.: A typology of identity-related crime: Conceptual, technical, and legal issues. Information Polity (2008) (forthcoming)

[KvGtH$^+$04] Koorn, R., van Gils, H., ter Hart, J., Overbeek, P., Tellegen, P., Borking, J.: Privacy enhancing technologies, whitepaper for decision-makers (2004)

[Lee08]   Leenes, R.: User-controlled identity management as an indispensable tool for privacy protection. Int. J. Intellectual Property Management 2, 345–371 (2008)

[Les99]   Lessig, L.: Code and Other Laws of Cyberspace. Basic Books, U.S (1999)

[LGM$^+$05] Leenes, R., Graux, H., Meints, M., Rost, M., Zuccato, A., Delaitre, S., Maghiros, I., Nikova, S., Clauß, S., Andronikou, V., Kursawe, K., Geradts, Z., Koops, B.-J.: D5.2b: Id-related crime: Towards a common ground for interdisciplinary research. Technical report, FIDIS Consortium (2005)

[LT02]   Lenk, K., Traunmüller, R.: Electronic Government: Where Are We Heading? In: Traunmüller, R., Lenk, K. (eds.) EGOV 2002. LNCS, vol. 2456, pp. 1–9. Springer, Heidelberg (2002)

[Lyo01]   Lyon, D.: Surveillance society; Monitoring everyday life. Issues in society. Open University Press, Buckingham (2001)

[Lyo04]   Lyon, D.: Globalizing surveillance; comparative and sociological perspectives. International Sociology 19(2), 135–149 (2004)

[Mal06]	Malinen, J.: Windows cardspace, p. 7 (2006)
[Mar03]	Margulis, S.T.: Privacy as a social issue and behavioural concept. Journal of Social Issues 59(2), 243–261 (2003)
[MWB⁺04]	Mitchison, N., Wilikens, M., Breitenbach, L., Urry, R., Portesi, S.: Identity theft, a discussion paper, p. 33 (2004)
[OL08]	Oomen, I., Leenes, R.: The prime survey — a study regarding privacy attitudes and behaviour of students in the netherlands, flanders and the uk. Technical report, PRIME Consortium, May 31 (2008), https://www.prime-project.eu/
[Oll04]	Ollman, G.: The phishing guide, understanding & preventing phishing attacks, p. 42 (2004)
[OMS⁺07]	Olsen, T., Mahler, T., Seddon, C., Cooper, V., Williams, S., Valdes, M., Arias, S.M.: Privacy & identity management. Technical report, Senter for rettsinformatikk (2007)
[Opp04]	Oppliger, R.: Microsoft.net passport and identity management. Information Security Technical Report 9(1), 26–34 (2004)
[O'R05]	O'Reilly, T.: What is web 2.0; design patterns and business models for the next generation of software (September 30, 2005)
[Org08]	The Gallup Organization. Data protection in the European union, citizens' perceptions (2008)
[otEC07]	Commission of the European Communities. Communication from the commission to the european parliament and the council on the follow-up of the work programme for the better implementation of the data protection directive (com(2007) 87 final) (2007)
[Par02]	Article 29 Working Party. Opinion 2/2002 on the use of unique identifiers in telecommunication terminal equipments: the example of ipv6 (wp 58), Technical report (May 30, 2002)
[Par06]	Article 29 Working Party. Opinion 8/2006 on the review of the regulatory framework for electronic communications and services, with focus on the eprivacy directive (wp126). Technical report (September 26, 2006)
[Par07]	Article 29 Working Party. Opinion 4/2007 on concept of personal data (wp 136), Technical report (June 20, 2007)
[Pat03]	Pato, J.: Identity management: Setting context, p.5 (2003)
[Pet02]	Petrovic, O.: The Immanent Fields of Tension Associated with e-Government. In: Traunmüller, R., Lenk, K. (eds.) EGOV 2002. LNCS, vol. 2456, pp. 187–190. Springer, Heidelberg (2002)
[Pfi03]	Pfitzmann, B.: Privacy in enterprise identity federation: Policies for liberty single signon (2003)
[Pis01]	Pisello, T.: Return on investment for information technology providers — using ROI as a selling management tool. Information Economics Press, New Canaan Conneticut (2001)
[PJBR06]	Paine, C.B., Joinson, A.N., Buchanan, T., Reips, U.-D.: Privacy and self-disclosure online. In: British Psychological Society Annual Conference, Cardiff (2006)
[PK03]	Patrick, A.S., Kenny, S.: From Privacy Legislation to Interface Design: Implementing Information Privacy in Human-Computer Interactions. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 107–124. Springer, Heidelberg (2003)

[PM03]   Pashalidis, A., Mitchell, C.: A taxonomy of single sign-on systems (2003)

[PN07]   Pitkänen, O., Niemelä, M.: Privacy and data protection in emerging rfid-applications (March 14, 2007)

[PR07]   Powell, A., Recordon, D.: Openid: Decentralised single sign-on for the web. Ariadne (51) (2007)

[PRI06a]   PRIME WP14.1. Framework v2. Leenes, R., Fischer-Hübner, S. (eds.) PRIME Deliverables, no. D14.1.b (July 2006), https://www.prime-project.eu/prime_products/reports/

[Pri06b]   Prins, C.: Property and privacy: European perspectives and the commodification of our identity. In: Guibault, L., Hugenholtz, P.B. (eds.) The Future of the Public Domain, pp. 223–257. Kluwer Law International, Dordrecht (2006)

[Pri07a]   Prins, C.: Electronic government: Variations on a concept. In: Prins, J.E.J. (ed.) Designing e-Government, 2nd edn. Kluwer Law International, Alphen aan den Rijn (2007)

[Pri07b]   Prins, C.: National perspectives on e-government and required regulatory change. In: Prins, J.E.J. (ed.) Designing e-Government, 2nd edn. Kluwer Law International, Alphen aan den Rijn (2007)

[PRI08]   PRIME WP14.1. Framework v3. Fischer-Hübner, S., Hedbom, H. (eds.) PRIME Deliverables, no. D14.1.c (March 2008), https://www.prime-project.eu/prime_products/reports/

[Pro06]   The Surveillance Project. Global privacy of data: International survey. Summary report, Queens University - Ipsos Reid (November 13, 2006)

[Raa04]   Raab, C.D.: Privacy issues as limits to access. In: Aichholzer, G., Burkert, H. (eds.) Public Sector Information in the Digital Age. Edward Elgar, Cheltenham (2004)

[Raa05]   Raab, C.D.: Perspectives on 'personal identity'. BT Technology Journal 23 (2005)

[Rac75]   Rachels, J.: Why privacy is important. Philosophy and Public affairs, 323–333 (1975)

[Re03]   Rogers, E.M. (ed.): Diffusion of Innovations. Free Press, New York (2003)

[Reg95]   Regan, P.M.: Legislating Privacy; Technology, Social Values, and Public Policy. The University of North Carolina Press (1995)

[RFI08]   Working Group RFID. Internet of things in 2020. Technical report, ETP EPoSS (May 19, 2008)

[RR04]   Rivera, M.A., Rogers, E.M.: Evaluating public sector innovation in networks: extending the reach of the national cancer institute's web based health communication intervention research initiative. The Innovation Journal: The public Sector Innovation Journal 9, 1–5 (2004)

[Sam07]   Samuelson Law, Technology & Public Policy Clinic. Security breach notification laws: Views from chief security officers. Technical report, Samuelson Law, Technology & Public Policy Clinic (2007)

[SAS06]   Sonnenreich, W., Albanese, J., Stout, B.: Return on security investment (rosi) — a practical approach. Journal of Research and Practise in Information Technology 38(1) (2006)

[Sch68]   Schwartz, B.: The social psychology of privacy. The American Journal of Sociology 73(6), 741–752 (1968)

[Sch92]   Schoeman, F.D.: Privacy and social freedom. Cambridge Studies in Philosophy and Public Policy. Cambridge University Press, Cambridge (1992)

[Sho03]   Shostack, A.: People won't pay for privacy, reconsidered, p. 5 (2003)

[Sil01]   Silcock, R.: What is e-government. Parliamentary Affairs 54(1) (2001)

[Soc07]   Wave 66.3 TNS Opinion & Social. E-communications household survey (2007)

[Sol07]   Solove, D.J.: The Future of reputation, gossip, rumor, and privacy on the internet. Yale University Press, New Haven (2007)

[Sri06]   Srivastava, L.: Pervasive, ambient, ubiquitous: the magic of radio (presentation) (2006)

[Sta02]   Stalder, F.: The failure of privacy enhancing technologies (pets) and the voiding of privacy. Sociological Research Online 7(2) (2002)

[Tay02]   Taylor, C.R.: Private demands and demands for privacy: Dynamic pricing and the market for customer information, p. 31 (2002)

[TR]   Tung, L.L., Rieck, O.: Adoption of electronic government services among business organisations in singapore. Journal of Strategic Information Systems 14, 417–440 (2005)

[Uni08]   Universal McCann. Power to the people. social media tracker wave 3. Technical report, Universal McCann (March 2008)

[VM01]   Vandecasteele, J., Moerland, L.: Groeimodel voor iv-functie - het systematisch weergeven van een herinrichtingsproces. Technical report, KPMG Management Consulting (December 2, 2001)

[Wac]   Wacks, R.: Personal information, privacy and the law, Oxford

[Wes67]   Westin, A.: Privacy and Freedom. Atheneum, New York (1967)

[Wes04]   Westen, P.: The Logic of Consent: The Diversity and Deceptiveness of Consent As a Defense to Criminal Conduct. Ashgate, Aldershot/Burlington (2004)

[WF08]   White, M.D., Fisher, C.: Assessing our knowledge of identity theft: The challenges to effective prevention and control efforts. Criminal Justice Policy Review 19(3) (2008)

[WG99]   Wellman, B., Gulia, M.: Net surfers don't ride alone: virtual communities as communities. In: Kollock, P., Smith, M. (eds.) Communities and Cyberspace. Routledge, New York (1999)

[Whi04]   Whitman, J.Q.: The two western cultures of privacy: Dignity versus liberty. Yale Law Journal (113), 1151–1221 (2004)

[WP008]   WP03.1. 'F' series of deliverables – business processes and business case. In Ribbers, P. (ed.) 'F' Series of Deliverables – Business Processes and Business Case, number D2.2.a in PRIME Deliverables (May 2008), https://www.prime-project.eu/prime_products/reports/reqs/

[WP205]   FIDIS WP2. D 2.1: Inventory of topics and clusters. Technical report, FIDIS Consortium (September 21, 2005)

[Ze73]   Zaltman, G. (ed.): Innovations and organisations. John Wiley & Sons Inc., New York (1973)

Part III

What Technology Can Do for Privacy and How

Introduction: Privacy, Trust, and Identity Management

Stephen Crane[1], Siani Pearson[1], and Dieter Sommer[2]

[1] HP Labs
[2] IBM Research

Part III of the PRIME Book is dedicated to an in-depth discussion of the technological side of the PRIME project. Part III is structured such that the current chapter gives an introduction to this part of the book and the concept of trust which is foundational to PRIME and its choices of technology. The technical discussions start with a discussion on architectural aspects with the focus on a privacy-enhancing architecture describing how different privacy technologies can be integrated into a system for an improved protection of the privacy of users in Chapter 9. The discussions on the architectural aspects are followed with elaborations on relevant technologies that are described in detail following the chapter on the architecture aspects. The detailed structure of Part III is presented further below in Section 8.2.

PRIME technology and its architecture have been built with multiple key goals in mind in order to improve the protection of the data privacy of users:

Establishment of trust: PRIME technology allows two parties who engage in an interaction to establish mutual trust. Trust in the other party in a specific situation means that one is assured that the other party will behave as expected. This is mainly achieved by the mutual exchange of data which allows for a better assessment of the respective other party. This process of establishing trust requires the interplay of multiple technologies in a strongly orchestrated way and forms the backbone of the PRIME Architecture. The 'source' of the data to be used in the trust establishment process is generally not restricted in PRIME. Examples are common certified attributes, assurance data about a service provider, reputation data, and platform integrity metrics, to name some important kinds of data.

Reduction of trust requirements: One of the key goals of PRIME has always been a relaxation of the strong trust requirements in terms of

proper handling of their data of users in other parties like service providers and certifiers. This is particularly relevant in the context of releasing data: Today's way of interactions in communication networks typically makes a user linkable over all her interactions and thus both service providers and identity providers and other parties such as content aggregators need to be fully trusted to handle their data appropriately, particularly to not jointly establish extensive user profiles by pooling their transaction logs. We use anonymous credential system technology to allow for privacy-enhancing exchange of certified user data in a model of reduced trust requirements in the players of the system.

Data minimization: PRIME strives at implementing the concept of data minimization for interactions of users with other parties. Data minimization means that a party needs to release only the data strictly necessary that the other party can provide the requested service. Data minimization is closely related to the two goals discussed above: It is applicable in the protocols for data exchange between parties, that is, in the protocols for establishing trust between parties. The reduction of trust requirements is an important means of also obtaining more data minimizing systems for the release of data, for example anonymous credential schemes.

Automated policy enforcement: In all interactions when data are released, data handling policies are agreed between the parties defining how the data should be handled. The data handling policies need to be enforced once the data of users are held by service providers. Data handling policies require access control mechanisms on the one hand and privacy obligtation enforcement mechanisms on the other hand. Both are accounted for in PRIME in order to perform an appropriate enforcement of policies after data have been released.

A substantial fraction of the technological part of PRIME treats the problem of how two parties can establish mutual trust among each other while at the same time not compromising their privacy. Trust is one of the core concepts that is relevant throughout the book, and particular this part, and is thus discussed next to give the reader a better impression of the meaning and different flavours of trust.

8.1   Trust

This section discusses the concept of trust which is a fundamental concept of the work within PRIME. Our architecture addresses parts of those trust aspects, others are out of the scope of what can be addressed with technology. The main aspects the architecture addresses are the assessment of trust of other parties through technological means.

8.1.1 Analysis of Trust

To date, we have no universally accepted scholarly definition of trust. Evidence from a contemporary, cross-disciplinary collection of scholarly writing suggests that a widely held definition of trust is as follows [DRC98]:

> Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviour of another.

Yet this definition does not fully capture the dynamic and varied subtleties considered below.

Approaches to modelling trust in social science include:

- **Temporal aspect.** Trust has been considered to have a temporal aspect for a long time, ever since Aristotle stressed that friendship cannot exist without trust and that trust needs time. In the twentieth century, Niklas Luhmann viewed trust as a representation of the future. This is rather similar to the belief we hold when reasoning inductively that after experiencing a historical pattern of behaviour, similar behaviour can be expected in the future. In the personal sphere, trust is a historical process of individuals learning to trust others without having to give unlimited trust. However, according to [Luh79], we do not really understand the process.
- **Risk aspect.** Social scientists have strongly stressed that risk is a central aspect of trust. For example, Luhmann believed that trust is an investment that involves risky preliminary outlay, where we accept risk in order to reduce the complexity of what we think about the world, in order that we may function [Luh79]. In a similar vein, Georg Simmel believed that trust is an intermediary state between ignorance and knowledge, and the objective of gaining trust may fail [Sim68]. Again, more recently, Nissenbaum in [Nis99] stressed how trust involves vulnerability.
- **Delegation.** One reason why trust is necessary is because we do not have the resources on a personal level to analyze all the information that we need during our working life. Therefore, as societies become more advanced, social order is replaced by legal order and delegation increasingly requires trust in functional authorities and institutions, particularly in the area of knowledge (and technology). However, if these institutions or powerful individuals let down the people who trust them, there is the risk of a big change of attitude towards them. This leads us to the following point:
- **Dynamic aspect.** There can be differing phases in a relationship such as building trust, a stable trust relationship and declining trust. Trust can be lost quickly: as Nielsen states [Nie99]: "It [trust] is hard to build and easy to lose: a single violation of trust can destroy years of slowly accumulated credibility".

Further analysis of online trust, and of how trust may be underpinned by specific technological mechanisms, is a key issue of multiple PRIME technologies. The interested reader will find trust aspects, particularly how to build trust between parties, in the remainder of this part of the book. Keep in mind that there are different flavours of trust when reading the contributions.

8.1.2   Establishing Trust and Managing Privacy

Being able to say that another party can be completely trusted to handle personal information with today's technology is probably unrealistic. Unless we can 1) completely isolate the processing from the operator and 2) rely on the technology and implementation, we have to rely on some level of faith in the other party. Requirement 1) is unrealistic since in practice virtually every application is likely to involve some form of human intervention, including access to the information after the 'trusted' processing is complete. Requirement 2) is currently difficult to demonstrate. Since in practice we are unable to prove 'before the event' that a recipient is trustworthy and will uphold a user's wishes, the next best approach (as in real life) is to establish an alternative means of enforcement. A contract provides a user with an indication that a recipient intends to carry out the user's wishes and is a means to identify deviation from agreed actions. Of course, the contract is only useful if it is enforceable. A deceitful recipient will most likely always be able to circumvent controls. However, the concept of a contract is useful for a recipient who has every intention of behaving properly, and wishes to demonstrate so in order to differentiate themselves from other less scrupulous recipients. To some extent this lessens the enforcement challenge, making it an obligation of the recipient. For the most part these large corporate organisations have a strong brand (which itself can be a basis for trust) and generally intend to behave honourably and fairly. Often the later is enforced through third party legislation and codes of conduct. These are the organisations that are willing to demonstrate the openness of their procedures and be held accountable for misconduct.

8.1.3   Understanding Trust

As already discussed, trust is a combination of social trust and technical trust. Both of these aspects of trust influence the user's overall trust assessment. Another way to look at trust is in terms of the three components: technical (as before), history and reputation. (Some may consider history and reputation to be the same, however there is a subtle difference.) History and Reputation form the social assessment, and each is based on past interaction with the intended recipient. In the case of history the assessment is made on past interactions that the user has had. Reputation is based on interactions that others have had. Reputation introduces a further complexity in that the user

also has to judge the trustworthiness (or reliability) of the third party's assessment. The user must also be aware that the quality of a reputation indicator may vary depending on the provider, and be ready to compensate.

Reputation is clearly strongly influenced by social understand, but history (as perceived by the user) is measurable as long as the user can articulate the conditions under which past performance has a bearing on future performance. It is this ability of the user to collect and assess evidence that is directly related to past events that provides a means to form an opinion about trustworthiness in the absence of other more definitive trust indicators.

Users use the following criteria to establish that an organisation is trustworthy:

1. Acceptance by the organisation of agreed terms and conditions that describe how the PII can be used.
2. Endorsement of a organisations' privacy compliance by a TTP, e.g. external auditors, privacy seal
3. An organisation's willingness to communicate the status of a user's PII data to the user whenever the status of the data changes, e.g. at the time when the data is deleted.
4. The user's ability to interrogate the organisation at any time and check the status of their PII data.
5. Past performance. This only becomes relevant for subsequent interactions. The status of previous or outstanding interactions should influence trustworthiness. Probably the best way to handle this is to conclude that any non-compliance raises reason for concern.

Even so, there will be situations where not all factors are met, and the user must decide how to proceed. Trust alone may not be the only influencing factor, and users may also consider context, availability of other options and risk vs. gain. PRIME has developed and integrated technologies that help a user assess the trustworthiness of a service provider.

8.1.3.1   Trustworthiness of Services-Side System

Knowing that an organisation has adopted state-of-the-art trust technologies can be an initial sign to the user that the organisation intends to be true to their word. Today, state-of-the-art trust technologies include a TPM (Trusted Processing Module) that provides:

- A reliable third party endorsed stable identity
- Originator non-repudiation achieved through TPM-controlled signatures

These requirements can be achieved by equipping a server with a TPM, endorsed by a Trusted Third Party, and building the functionality to allow 1) remote interrogation of the TPM by the user, and 2) automatic signing of acknowledgements and other information intended to convince the user that their wishes are being fulfilled. In practice, the systems that support services

offered by an organisation will be much more complex than a simple peer-to-peer arrangement. Whilst these systems may be built on TPM and future trusted platform technologies, techniques for forming an aggregated measure of trust across multiple heterogeneous systems that process personal information still need to be researched.

8.1.3.2   Trustworthiness of the Organisation

Trust in an organisation is built up over time, based in part on past interactions. An initial assessment of an organisation can be done using PRIME technology. Evidence that an organisation is willing to commit to an intended action, possibly in the knowledge that to not do so will incur penalties, is a useful sign of good intentions.

Typically, the user would either review or present the terms under which the interaction will take place (i.e. a policy or contract). Once accepted, these terms are binding to some degree. As required, the user reviews the interaction and compares outcome against the contract, particularly where the terms specify several points in the process where an assessment can be made (c.f. project milestones). This leads us to a process with clearly definable steps:

- Policy/contract comparison between user and organisation
- Fulfilment (by the organisation)
- Checking (by the user)
- Opinion forming (by the user – essentially retention of evidence to aid trust evaluation during future interactions.)

The proposed approach differs from existing approaches (e.g. P3P) by providing feedback to the user and indeed involves the user / user's system in the process of 'active' comparison and management.

8.1.3.3   User-Side Trustworthiness

Whilst the user is concerned about the trustworthiness of the services provider, the user must also be able to trust their own system to hold their personal information securely. Assuming that the user is the only person with legitimate access to the system, trust is based solely on the technical merits of the system. Again, taking the TPM as the state-of-the-art technical security solution, the functionality to be supported by the TPM should include:

- Granting user authorised access to personal information, i.e. identification and authentication of the user.
- Secure storage of personal information and/or the cryptographic key(s) used to control access to personal information.
- Generation of random 'seeds'.

Additionally, the TPM permits the generation/presentation of pseudonymous identities that may support or supplement credential management schemes like DRIM [Tec] and Identity Mixer [CL01a]. Many users are likely to find the task of managing trust too difficult because it requires specialist skill and knowledge. Ways of providing help and support to the user through UIs, warning mechanism, best practice advice, etc. will need to be deployed to help users check/preserve their platform's trustworthiness and avoid making decisions that could compromise their platform. These are ambitious goals, involving long-term research, but we can start by leveraging the functionalities provided by TPMs and trusted platforms. Looking further into the future, and the evolution of ambient services and devices, managing trust on the user-side goes beyond the relatively straightforward 'gatekeeper' role that we see here to that of an 'agent'. Imagine the situation where a user has a need for particular service, and instructs their personal system to 'look' for the most appropriate services on offer. Part of this process could involve the automatic release of personal information about the user. How can the user be confident that their personal system is acting in the best way to preserve their privacy? By concentrating on the specific situation described, i.e. where the organisation is essentially trustworthy but needs to be able to demonstrate this publicly, we can provide users with the means to differentiate likely trustworthy from untrustworthy parties to which the user intends to release personal information.

8.2   Structure

Part III is structured in the following way: After this introduction, we present an overview of the PRIME Architecture in Chapter 9 give the big picture of the PRIME technology and the interplay of the various mechanisms in a single system. The main contributions of the architecture chapter is to integrate multiple of our privacy-enhancing technologies into a single system. Particularly, this includes the definition of a data representation which is a basic prerequisite for an integration of our technologies, generalizations to mechanims we use in the architecture, and a negotiation protocol for allowing two parties to establish trustworthiness through the exchange of data. The architecture goes into deep technical details in its focus areas while it remains abstract in the areas that are covered already in the other chapters of Part III.

The architecture is followed by chapters dedicated to the individual concepts and technologies PRIME is based on. Each of those chapters contains a detailed discussion of the technology or concept at hand. We note that already a subset of those technologies is sufficient to build a basic privacy architecture. Further technologies can be added on top to increase the scope of protection.

Chapter 10 discusses anonymous credential systems, that is, cryptographic mechanisms for the privacy-friendly exchange of certified data. Such systems

are a powerful tool for allowing a service provider, or other user in a peer-to-peer setting, to establish trust in a user based on attribute information certified by one or more third parties, such as a government. Such a system allows the user to release data in a very much controlled way, as required for the ongoing interaction. Particularly, third parties such as identity providers do not learn interaction histories of the user which is a major problem in traditional such systems. The Identity Mixer anonymous credential system comprises one core component of the trust establishment process of PRIME.

In order to make use of anonymous credential systems and other systems for revealing (certified) information, such a system must be embedded into an elaborate policy-driven system for governing the request and release decisions of the involved parties. Chapter 11 discusses PRIME's access control and data handling models and languages. Those models and languages are designed to serve multiple purposes: 1) Protecting resources of a party, where resources can be customer data or the party's personal data; 2) agreeing data handling policies and enforcing the access control aspect of such; and 3) providing authorization decisions for the trust establishment process between two interacting parties.

The handling of privacy obligations is, to a large extent, complementary to access control. Privacy obligations are one integral part of the data handling policy agreed between a user and a service provider or another user before personal data are actually released by the user. They define actions to be executed on specific data items once certain conditions are fulfilled. This can comprise actions such as deleting data after a certain retention time has passed or encrypting data with an archive key once the data are not used any more in the operational database. Chapter 12 discusses the model and language aspects of the privacy obligation management approach of PRIME.

While the previous two chapters 11 and 12 discuss the models and languages for access control, data handling, and obligations, the next two chapters discuss the systems aspects of implementing such models in practice. The implementation aspects of access control and data handling are discussed in detail in Chapter 14. This chapter particularly elaborates on some interesting architecture aspects that deserve consideration. The system implementing the model for privacy obligation management is detailed in Chapter 15.

Chapter 13 goes into detail in the model and language for handling assurances within PRIME. Assurances are a special category of data statements provided by service providers to users in order to increase user trust in the service providers. Assurance policies and access control policies both have an impact on the trust establishment process in an interaction between two parties and are, considered from this point of view, conceptually related. Chapter 16 discusses system aspects of the assurance model and language and the relation to the PRIME Architecture. A specific part of the assurance aspects—the assessment of the trustworthiness of platforms—is covered in Chapter 17.

Chapter 18 gives insight on various privacy-enhancing mechanisms research of which has deserved some attention within the PRIME project.

Those mechanisms comprise: Privacy measures, data anonymization, anonymous communication, and unobservable content access. The section on privacy measures deals with privacy metrics that allow to express the degree of privacy a user has in an interaction. Of those, anonymous communication is a core mechanisms used in the PRIME architecture and prototypes for achieving our data minimization goals by not revealing users' network addresses by default.

Despite certified attributes from reputable identity providers such as governments are a useful means for establishing trust once approriate identities will be widely available, there exists another powerful way of establishing trust: Reputation. Reputation allows for building trust not based on the statement of one of a few (trusted) entities, but rather on the knowledge, experience, and perception of a large set of people, e.g,. users of a service. Chapter 19 discusses the topic of reputation mainly from the perspective of general overview and architecture.

Chapter 20 discusses user interfaces, a field of research without which security technology is hard to impossible to delploy successfully. The focal points of the user interface work discussed are usability, security, and legal compliance. All of those aspects must be met for a user interface meeting practical expections by the various stakeholders. From a functionality point of view, multiple impmortant functions of user interfaces for privacy-enhancing identity management are covered: attribute selection, which is closely related to anonymous credential systems for providing the selected attributes; display of privacy policies; trust and assurance assessement of the other party; and data tracking. Getting user interfaces right is a key requirement for an identity management system working in the user-centric model envisioned by PRIME.

Following the discussions on technical mechanisms so far in this part of the book, the authors discuss technology assurance in Chapter 21. In a nutshell, technology assurance deals with processes for ensuring a certain level of quality of software, components, or systems. In such a process the software gets evaluated against a standard set of criteria. The chapter particularly dicusses the approach of PRIME of an early evaluation already during the development process with the goal of reducing the overall cost of fixing problems.

Part III of the book finally closes with a discussion of multilateral interactions, that is, interactions involving potentially more than two parties, in Chapter 22. We think that such interactions are of particular interest in the near- to mid-term future because such interactions are becoming increasingly prominent in the online world, e.g., in the space of user-generated content, online collaborations, or social networks. The discussions on this subject are less technical and geared towards giving an insight into the requirements for multilateral interactions based on PRIME scenarios rather than proposing solutions. This way, the chapter provides an outlook to future work extending the scope of the core aspects of the PRIME project.

9

Architecture

Dieter Sommer

IBM Research

9.1 Introduction

This chapter introduces an architecture for privacy-enhancing identity management. The architecture can be used as a blueprint for building a comprehensive system with a plurality of players for privacy-enhancing identity management. It elaborates on how to integrate state-of-the-art privacy-enhancing technologies (PETs) to achieve the goals as outlined below. Important concepts realized by our architecture have been put forth in the European data protection regulation, particularly the European Data Protection Directive [Eur95] and its implementations in the EC member states' data protection laws. Those foundational concepts include data minimization, data quality, transparency, the finality principle, and subject access to data.

The structure of this introduction is as follows: We give an overview of the most severe weaknesses in today's Web in terms of identity management next, and, based on this, motivate main goals for our architecture. Then we outline the technologies we attempt to reach the goals with. This is followed by an overview of state of the art. We close the introduction with an outline for this book chapter.

9.1.1 Motivation and Goals

As a motivation for our work, we give a brief overview of the situation of identity management support for end users in today's Web. The need for identity management support for end users is given whenever a user interacts with another party, typically a service provider, over an electronic communication medium. In today's Web, when a user engages in an interaction with a service

provider, there is only suboptimal support available for the user in handling the identity-related aspects of the interaction. This holds, for example, for *assessing the potential trustworthiness* of the other party; she needs to manually assess information, such as privacy seals presented on the Web page, that might support her trust evaluation of the other party. When it comes to the *assessment of the privacy policy* of the service provider, the user must read and interpret it herself in order to make a conscious decision on continuing the interaction. This can be tedious, time consuming, and prone to interpretation errors, unless the privacy policy is written in a very concise, clear, and understandable way, which is, for many service providers, not the case today. Regarding the actual *release of (personal) data*, there are multiple problems in today's Web architecture as well: in most cases, data are requested by a service provider through a Web form and provided in plaintext form by the user through entering them into the form. The only widely-deployed user support here are form fillers, e.g., the ones included in the major Web browsers, to take the tedium off the user of repeatedly entering the same attribute data. This prominent way of handling personal data in Web interactions today has not been designed for handling certified user data. That is, data are uncertified, meaning that often times more data need to be requested than actually required by the business process in order to cross-check the required data for correctness. This use of uncertified data is one reason for the excessive release of personal data which is one of the most pressing problems of identity management in today's Web. The protocols that are upcoming in practice for the release of certified data suffer from severe privacy problems of excessive data release, either to the data recipient or to the certifying party or both. Furthermore, such protocols are not widely used as of today, only for few or closed special-purpose applications, such as e-Government. Automated solutions that *enforce the privacy policy* promised to the user are not widely used today by service providers, thus (unintentional) violations of the agreed policies for handling data are a problem. This is particularly problematic when data are further released to other parties by a service provider without the agreed data handling policy remaining associated with the data. A general property of the current Web architecture and the way the interaction with users regarding personal data is implemented is that the only user-side software is a standards-compliant Web browser which does not implement any advanced identity management functionality. This means that the *user interface is not consistent*, that is, it varies from service provider to service provider, and thus makes the situation difficult for the user compared with the situation of having the same user interface for identity management interactions with any service provider.

Considering the above-described situation of identity management in today's Web, we conclude that there is a strong need for a comprehensive architecture for identity management in the Web and electronic communication media at large. Such an architecture is an enabler of what is commonly referred to as *informational self-determination* of the individual. The

concept of informational self-determination refers to users having the possibility of consciously determining themselves what may and may not happen with their personal data, that is, to have control over their data. This particularly includes decisions on which parties to release what data to and what the conditions are for handling the released data. Today's identity management approach on the Web provides users, due to its very pragmatic approach and the lack of an overall architecture incorporating advanced mechanisms, insufficient support for their informational self-determination when considering what is technologically possible.

The central goal of our architecture is to allow users to better excercise their right for *informational self-determination* in electronic interactions. It thereby must provide a blueprint for a comprehensive identity management system that covers the life-cycle of identity data and addresses the problems we face as outlined above. The architecture must particularly address the aspects of allowing a user to assess the party she is interacting with, evaluate the privacy policy of this party, and allow for the release of certified attribute data, thereby precisely fulfilling the other party's data request without revealing any excessive data. The latter is crucial for achieving data minimization, the concept of reducing the amount of data released by a user to what is required for the interaction at hand, thus avoiding any excessive release of data. The architecture must allow for a consistent user interface to be displayed to the user, regardless of the service provider she is interacting with. The architecture must also go beyond what is done today in terms of automated enforcement of the privacy policy in the back end system of data recipients. The operation of the architecture should be driven by policies, that is, machine-interpretable rules expressed in a formal language, also on the user side, in order to implement as much automation of identity management actions as possible. This allows for a (semi-)automatic processing such that most of the effort is done by the machine as determined by policies. Regarding the trust model, an important objective is to reduce trust in other parties such as service provideres as much as possible in order to protect users from parties that want to learn more information about the user than they should be allowed to learn.

9.1.2   Realizing the Goals: Technology

In order to realize the above-mentioned goals, a choice of classes of technologies needs to be made. We summarize the classes of technologies our architecture is built on next: Access control policies dictate the requirements that another party has to fulfill in order to access a resource such as a service or data and authorization systems evaluate such policies and compute an access decision. Data handling policies specify how data are to be handled by any recipient. Private certificate systems (anonymous credential systems) allow one to reveal certified data in a controlled way without revealing excessive data. Negotiation protocols drive the mutual data exchange and agreement on data handling policies between two interacting parties, allowing a user to

request data from a service provider for an assessment of it and allowing a service provider to request data from a user to be able to grant the user access to the service. Life-cycle data management systems enforce the agreed data handling policies in the back-end system of a service provider once data are released. A user-side user interface consistent throughout service providers involves the user into the interactions related to identity management and allows her to configure her system. In the process of building the architecture, well-suited technologies from the mentioned classes have been selected and tightly integrated with each other, thereby obtaining our privacy-enhanced identity management architecture bringing us closer towards our goal of informational self-determination of the individual. Next, we give more details on concrete technologies and concepts we build upon.

Negotiation: A central function of our architecture is what we call *negotiation*. A negotiation is a mutual exchange of data between parties and agreement on data handling policies for those data. A negotiation is realized through the negotiation protocol, a protocol between two parties, e.g., a user and a service provider. A negotiation is triggered whenever a party accesses a resource of another party, e.g., when a user accesses a service offered by a service provider by clicking a link on its Web site. A negotiation can comprise multiple rounds of data requests and responses by either party. Within a negotiation, the service provider obtains the data they need from the user in order to provide the service, and the user obtains the data from the service provider that they need to assess the service provider. The data requests in a negotiation are, amongst others, determined by the access control policies of the involved parties.

Attribute-based access control: Access control policies determine the data requirements a party has with respect to another party when the other party intends to access a resource of the party. Access control policies are specified on services and data, particularly also a user's data. Our policy language is based on ideas of the language of Bonatti and Samarati [BS02b] supporting attribute-based access control. That is, access decisions are not done based on identifiers as in traditional access control systems, but rather on arbitrary attribute data of the requesters. The access control language is an important building block for realizing data minimization in that it allows for the definition of minimal data requests based on certified attribute information. The access control policies determine the data exchanges in a negotiation and thereby are an important technology for an automation of the interaction between a user and a service provider.

Automated data handling: Data handling policies determine how data to be released to a party will be handled by this party. Both a service provider and a user specify data handling policies – the service provider to express how it will handle received data, and the user to express how she wants her data to be handled, once released. During a negotiation, the involved

parties agree on data handling policies to be applied to the data to be released based on their respective policies.

Privacy-enhanced data exchange: Privacy-enhanced data exchange is concerned with the protocols for a party obtaining certified identities from a certifying party as well as releasing parts of certified identities to other parties, such as service providers. An identity in this context is a set of attributes. The prefix "privacy-enhancing" refers to better privacy properties that are achieved in contrast to traditional attribute exchange technologies, such as standard X.509-style attribute certificates. As our main technology, we employ private certificate systems (anonymous credential systems), particularly the Identity Mixer system of Camenisch and Lysyanskaya [CL03, CL04]. Such systems allow for unlinkability between the interaction for obtaining certified identity data and releasing parts of those as well as between multiple such releases.[1] They particularly allow for a subset of a certified identity being revealed, thus one can precisely satisfy a data request without any technology-dependent release of additional data as would be the case when, for example, using traditional attribute certificates. This feature is particularly interesting in combination with the attribute-based access control policies to achieve data-minimizing interactions. Private certificate systems offer the strongest privacy protection features among all practical technologies for attribute exchange. Those intrinsic properties of private certificate systems allow for strengthening the underlying trust model for protocols for exchanging attributes.

Data model: In order to represent identity data and metadata in our architecture and to make different technologies interoperable, we need a well-specified approach to modeling data throughout the architecture. We denote this as *data model*. The data model is used to represent requests for data and responses to those, to store data at a party, to express the identity-related parts of access control policies, or to express the input to the data release subsystem. The data model goes far beyond the often used approach of represening data as lists of attribute-value pairs. Our data model must be able to express more elaborate statements on data, e.g., inequalities between an attribute and a constant, or disjunctions of sub-statements. Those features are mandatory for realizing the concept of data minimization to its extreme. The data model acts as "glue" between different technologies as it is in many cases used as input or outout in invocations of services of components. This further underlines its importance for the architecture.

User interface: Our architecture defines a client-side system including a user interface realizing the following functionalities: presentation and

---

[1] Of course, the unlinkability between multiple releases can only hold in case the revealed data do not make the interactions linkable. To be more precise, the certification of the data does not lead to linkability as is the case with traditional signature schemes and thus attribute certificates.

adaptation of the data handling policy of the other party; presentation of identity data requests and fulfillment of those requests; presentation of data about the other party; editing of access control and data handling policies; and on-line access to data. The user interface is implemented by the client-side system and thus is the same regardless of the service provider the user is interacting with. The services-side user interface is mainly concerned with policy definition and editing and is not a focus of our work.

Data life-cycle management: Our view on data life-cycle management refers to the automated handling of data according to the agreed data handling policy once they have been released by the user. This encompasses the time-driven deletion or anonymization of data as well as the notification of the user in case of certain events. Concretely, we integrate with the framework of Casassa Mont of Chapters 12 and 15 which executes actions once specified events have occurred and a specified condition holds. The framework is extensible in terms of actions (workflows) that can be executed.

Considering the strong focus on the informational self-determination of users, the related goals, and the resulting technology choices for the architecture, it has been a clear choice that the architecture is designed to be *user centric* [BSCGS06]. Being user centric means that the user plays an active role in her identity-related interactions and thereby receives substantial control over her data. A party can decide to define policies on how their data should be handled such that the system can, on the user's behalf automatically execute those policies and enforce the intentions of the user. As mentioned, a main goal was to reduce the required trust in third parties wherever practical, that is, strengthen the trust model, by using advanced cryptographic protocols.

9.1.3   Related Work

At the time of the design of this architecture, no similarly powerful architecture addressing the problem space of privacy-enhancing identity management had been proposed in the literature or practice. Thus, our architecture can be seen as the first comprehensive effort towards a better implementation of the European tradition of privacy in tomorrow's data processing systems. Meanwhile, other initiatives have emerged, having goals similar to ours, particularly in the area of privacy-enhanced data release. A relevant upcoming architecture is the CardSpace architecture that has been proposed by a major industry player. It has lately started as well to investigate the use of private certificate technology for attribute exchange with stronger privacy properties than in traditional systems. Our architecture goes further in terms of not only considering privacy-enhancing data exchange, but also the assessment of the service provider through the user, the fully-automated enforcement of agreed privacy policies in the back-end of service providers, as well as including privacy policies and their processing. To this end, we have integrated multiple

technologies around a powerful data model that forms the common language the different components speak. A comprehensive overview of prior art and related work can be found in the individual chapters on PRIME's technologies following the architecture in the remainder of Part III of the book. For a discussion of legal and social aspects and other non-technical aspects of privacy-enhancing identity management, we refer the reader to Part II of this book and the given references.

The work on the architecture has been carried out related to the European PRIME project [PRIa] where PRIME is the acronym for "Privacy and Identity Management for Europe". The PRIME project had the goal of addressing the identity management challenge from an interdisciplinary standpoint comprising technical, legal, social, and economic aspects. The architecture was one of the main technical results of the project. It has evolved through four versions, V0 to V3 [Som04, Som05, CCS06, Som08] during the project, each of which is focusing on different aspects. V0 was the initial draft of the architecture and has been consolidated by V1. V0 and V1 were intended to have mainly the PRIME project participants as audience. V2 then took the approach of presenting the matters in the style of a reference architecture with a stronger focus on external audiences. V3 was the final version targeted at both internal and external audiences. The current book chapter is a consolidated write-up of the architecture and related ideas, considering also the main "lessons learnt" throughout and beyond the project. It focuses on a generalized data model to represent data of parties in a privacy-enhanced way and goes into technical details more than the different architecture documents. The focus from a content perspective is on privacy-enhancing data exchange based on private certificate systems and the required technology-related aspects. Technical as well as non-technical documents that have emerged from the PRIME Project can be found at [PRIa] and in a variety of conference and journal publications. The presentation of the architecture focuses on the parts that are not explained in the following chapters of the book which comprises the aspects of integrating the used technologies as well as extending them towards stronger privacy. The text incorporates improvements and generalizations of the originally-built architecture and thus may deviate from the following chapters describing the used technologies in terms of notation and generality.

Our main contribution is an architecture that specifies how to integrate multiple privacy-enhancing technologies and also how to orchestrate their use. A main part of the integration is a data model, that is, a formalism of representing data throughout the architecture. The data model is the common "language" spoken by different parties and components within the scope of a party. Based on the data model, we define the representation of different kinds of data a party needs to hold. With this foundation, we adapt existing technologies to fit together and extend their functionality in terms of privacy protection. We define a negotiation protocol for a mutual exchange of data and agreement on data handling policies based on the privacy-enhanced

authorization system we employ for determining access to resources such as services or data.

9.1.4   Outline

For this being instrumental for the understanding, we first provide an overview of the architecture, including the parties, their interactions, the different types of data being relevant for the architecture, and the components of a party's system in Section 9.2. In Section 9.3 on the data model we go into details on how data are represented in a formal calculus, while in Section 9.4 we apply the data model to the different kinds of data a party needs to handle. The sections just referred to are the conceptual backbone of the architecture as we repeatedly refer back to the concepts introduced there. Thereafter, we present one core piece of the architecture, concretely the part of the architecture related to privacy-enhanced data exchange based on private certificate systems, in Section 9.6. In Section 9.7 we discuss the underlying attribute-based authorization model and policy language and in Section 9.8 our framework for data handling policies. Based on our authorization system, we present in Section 9.9 our approach towards negotiation, that is, mutual release of data between interacting parties. We conclude the chapter in Section 9.10.

9.2   Architecture Overview

We present a high-level overview of our architecture in this section. This comprises the overall architecture model and the main components with their functionality. The overall focus of this presentation of the PRIME architecture is privacy-enhanced exchange of data between parties. The intention of this overview section is to provide the reader with the background and intuition upon which the remaining parts of this work build.

9.2.1   One Party in the System

We start the explanation of the overall architecture with showing which kinds of data and other items a generic party in the system, that is, a party independent of its concrete instantiation into, e.g., a user, service provider, or certifier, holds. Figure 9.1 illustrates these items a party holds and how they can be obtained.

First and foremost, a party holds *identity data* and *metadata* about itself and other parties. This comprises identifiers the party uses to refer to itself and other parties in the communication with other parties, possibly third-party-endorsed data about itself and other parties, records about data disclosures performed by the party, and data obtained about other parties. The stored data may be used locally by the party, e.g., the user or employees of a service provider to execute a business process, or they may be released to other parties. Data may be obtained by a party by being entered locally

through the console (user interface component) or other means, or obtained from other parties through the identity management system. Once data have been entered into the system at one party, the data are subject to being handled by the system in a policy-driven way until their deletion, entering and deletion thereby being the first and last phases in the life-cycle of the identity data. The architecture defines the management and protection of data from the initial to the terminal phase of the identity life-cycle with a focus on privacy-enhanced release of data.

A party also holds *policies* of various kinds: *Access control policies* specify which other parties, or persons or automated processes within the scope of a party, can get access to which data under what conditions; a part of access control policies is used for specifying the requirements the access requester must fulfill. *Data handling policies* specify how a party wants data to be handled (requirements of a party) how a party proposes to handle data (proposal of a party), and how data need to be handled by a recipient (agreed policy as result of a negotiation). The latter is the policy that needs to be enforced by a recipient of data and has been agreed in a policy negotiation protocol. Policies can be entered locally into the system, e.g., through a policy editor that is a part of the party's console, or obtained from third parties trusted for the purpose of providing policies, such as data or consumer protection organizations. Once a policy has been created or obtained by a party, it, or parts of it, may be communicated to other parties, e.g., together with data being released to the other party.

Ontologies are lists of rules expressed in a formal language and used by a party for making automated deductions over data or policies. For example, an ontology can define the concept of OECD_Government to be a government of an OECD country. For example, when a policy uses the concept of OECD_Government to specify the certifier of an identity statement, the party can infer, using automated reasoning based on this rule, that an identity statement endorsed by the German Government is an identity statement by an OECD Government. Ontologies can be defined by the party itself or again be obtained from third parties trusted for this purpose.

9.2.2 Parties and Interactions

The generic party explained further above captures what any party in the system can hold in terms of data and other items such as policies or ontologies. All parties are capable of performing the same identity management actions and interactions in our architecture which keeps the architecture flexible and conceptually simple. We call this property *party symmetric*. Though, depending on the tasks parties are commonly performing in a system, it makes sense that they be specialized into different *types* of parties. Important types of parties are users, service providers, certifiers, and conditional data recipients. Using this nomenclature to refer to different types of parties simplifies our
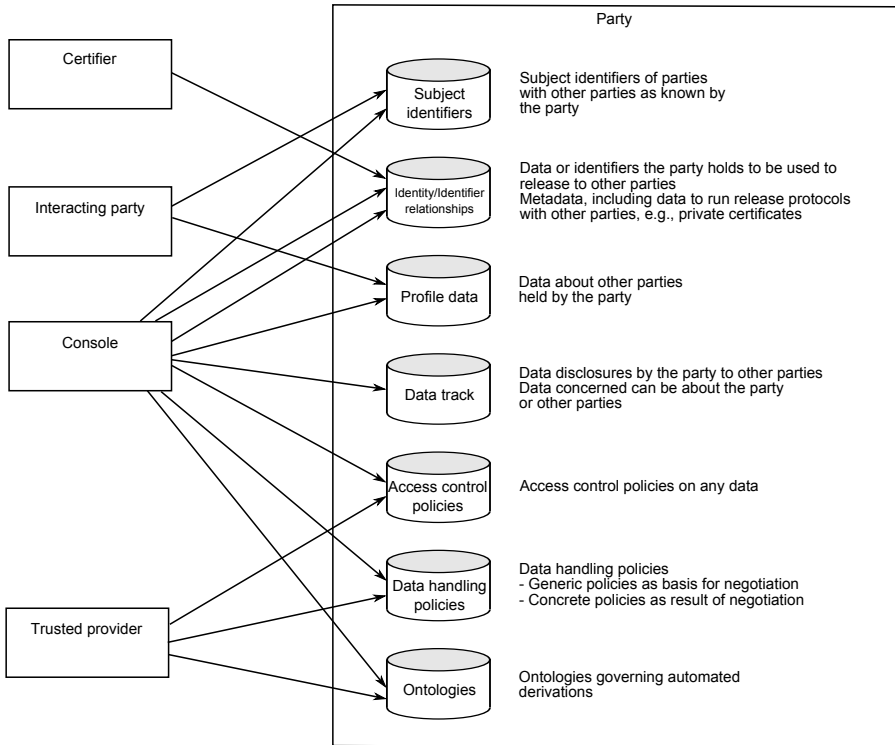
Fig. 9.1 Items held by a generic party

presentation of the architecture as we can refer to usual scenarios featuring those concrete parties instead of always talking about generic parties. The types of parties have differences in their concrete instantiation of the architecture due to different needs of the players. For example, from a user interface point of view, a user has a console that features easy-to-use interfaces targeted at the end user, a service provider or certifier has special-purpose administration consoles targeted at system administrators. In terms of availability, a large service provider needs to implement mechanisms such as load-balancing between multiple hardware platforms while a user's system does not implement such features.[2]

---

[2] Note that in our presentation, we assume that a user has a single system for her identity management tasks. We want to stress that a user may, in practical scenarios, have multiple systems, such as her office computer, her personal computer, and her smart phone. The data, policy, and ontology repositories (state) between those need to be kept synchronized by an appropriate approach. We do not further elaborate on this as this is an orthogonal problem but assume such a mechanism being in place. We describe the architecture from the perspective of each party having a single system as this is sufficient to capture our ideas.

9.2.2.1   Actions and Interactions

The parties can act and interact in a plurality of ways, each time acting under a specific role, as depicted in Figure 9.2. An action is thereby executed locally while an interaction is a protocol with another party. We next use a scenario with generic parties to present some important interactions in our system:

- $\mathcal{A}$ can establish an identifier relationship with $\mathcal{B}$ for creating an identifier about itself or an other party used to refer to such party in communication with $\mathcal{B}$.
- A party $\mathcal{A}$ can establish an identity relationship with $\mathcal{C}$, thereby obtaining the capability of later releasing parts of the contained data to other parties in a certified way.
- $\mathcal{A}$ can initiate a negotiation protocol with $\mathcal{B}$ by requesting a resource from the latter. During the execution of the negotiation protocol with $\mathcal{B}$, both $\mathcal{A}$ and $\mathcal{B}$ can release certified data to each other, depending on their authorization policy requirements. $\mathcal{B}$ requires attribute data about $\mathcal{A}$ in order to release the service while $\mathcal{A}$ requires information such as $\mathcal{B}$'s privacy practices and identity attributes.
- Party $\mathcal{B}$ can release data about $\mathcal{A}$ it has received in the negotiation to another party $\mathcal{D}$. Thereby, $\mathcal{B}$ may map the name it uses to refer to $\mathcal{B}$ to a new name used to refer to $\mathcal{B}$ in communication with $\mathcal{D}$ in order to break linkability between the different interactions.
- Party $\mathcal{B}$ can engage party $\mathcal{E}$ to revoke the anonymity of the transaction of $\mathcal{B}$ with $\mathcal{A}$ in case a pre-agreed condition becomes fulfilled, e.g., party $\mathcal{A}$ violating the agreed terms of service. Party $\mathcal{B}$ can enforce a policy agreed with party $\mathcal{A}$ on data previously released by the user to the service provider.

For any interaction where data are released, the involved parties agree on data handling policies to be applied to the data by the recipient. For such an agreement process, both involved parties input what they want or propose for handling the data, based on which the applicable policy is computed.

In such a generic scenario every party of a system could, in our model, be in any of the places, that is, act under any of the roles, and carry out the associated actions. Though, in practical identity management scenarios, a party usually has dedicated tasks and goals, according to who they are and what they want to achieve in the system. Thus, in such practical scenarios, a party will be specialized into a user, service provider, certifier, or other party according to the tasks it performs and goals it has.

The traditional interaction model between users and service providers is characterized by the key parties being users who are interested in consuming services offered by service providers. Service providers often require (certified) identity data about users in order to authorize them for access of the requested

service. Certifiers come into play for issuing identity relationships to users and service providers that allow them to release certified attribute data to each other. Once the service provider has obtained data from the user, the service provider can release (parts of) these data to another service provider as needed and allowed by the policies agreed between the service provider and the user. A conditional data recipient can be engaged by the service provider in order to de-anonymize transactions with a misbehaving user or realize other use cases in which the conditional data recipient can obtain data only once a certain condition gets fulfilled. As one can see, parties act under different roles in different interactions in such a scenario, for example the user is data provider and data recipient in a negotiation which equally holds for the service provider.
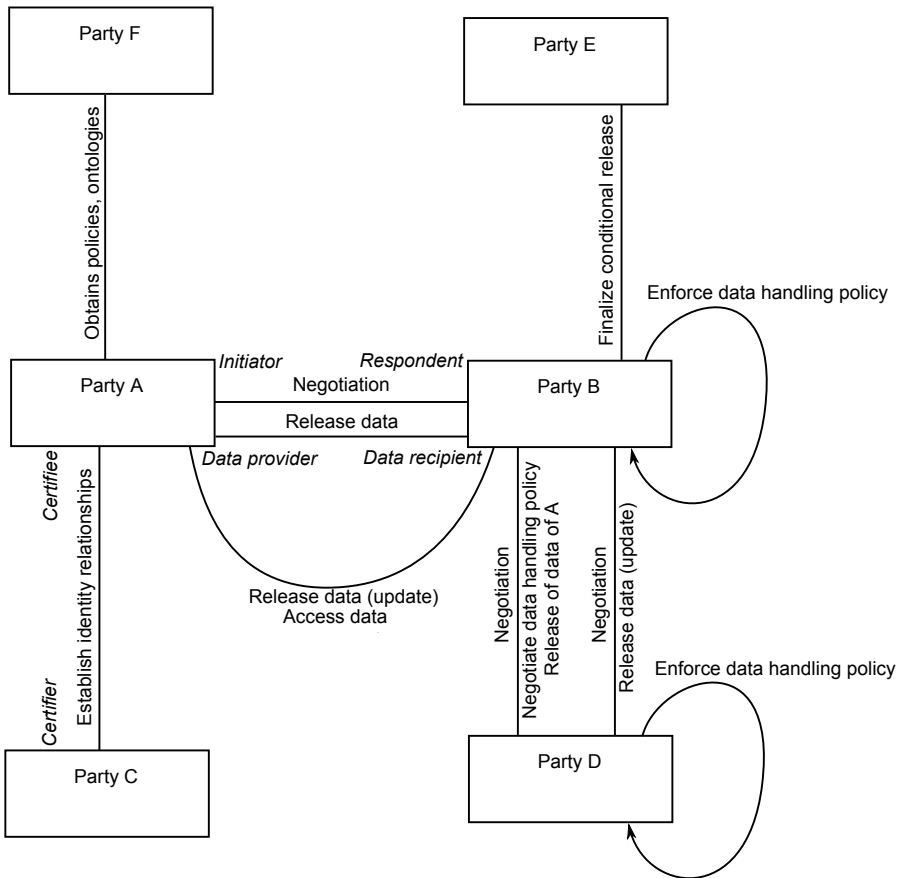
Fig. 9.2 Parties, their interactions, and associated roles

9.2.3   Data

As already mentioned earlier, our architecture deals with privacy-enhanced identity management which is centered around a single concept: *Data*. A party needs to, in order to take part in the identity management system, hold various kinds of data and uses statements about data to represent subject identifiers for parties used by the party, knowledge about a party's identity information that is vouched for by third parties, knowledge about which data have been released to which other parties, and knowledge about identity information about other parties.

When discussing data in an identity management context, we need to distinguish between *identity data* and related *metadata*. We want to note that it is hard or even impossible to come up with a formal definition of the concepts of identity data and metadata, though it is still useful to make the distinction informally for our discussions. We give the following characterization of the concepts: *Identity data* are data about attributes of parties in the system in a wide meaning. This comprises, e.g., a user's first name, or the predicate expressing that party $\mathcal{A}$ is of age greater than 18 years, or the statement that a party has passed the e-learning course on advanced accounting with a positive grade. Protection of data about parties, and particularly about users, is the primary concern of our architecture. Metadata, in general, are *data about data*. *Metadata* or *identity metadata* are data about identity data. This comprises data about the certifying party of identity data, data about the temporal validity period of a certification, data about when data have been released to another party, and so on. Metadata are required for various functions of the architecture, for example, for privacy-enhancing attribute exchange or for making trust decisions on identity data. The distinction between identity data and metadata does not only apply in the electronic world, but also for real-world credentials. Take as an example a (non-electronic) identity card. It contains identity data, namely the attributes first name, last name, citizenship, date of birth etc. In addition, it contains metadata over those identity data, such as the issuer (e.g., the German Government) and the validity period of the credential, that is, of the certification of the identity data contained therein.

In a privacy-enhancing identity management system a party needs to be able to release data about itself or others to other parties, keep track of its releases of data, and retain and use data about other parties. Data thereby is used for creating trust among parties, e.g., as certified attributes. Parties may appear under different identifiers (names) to other parties and parties may talk about other parties using multiple identifiers for those. This gives rise to the following "kinds" or classes of data held by a party: *Identifier relationships* the party holds, *identity relationships* of the party, the *data track* of the party, and *profile data* about other parties held by the party. Our motivation for splitting the data held by a party into those classes is the different purposes those data serve: Identifier relationships specify mappings between names of subjects and

are used by the party when referring to those subjects towards other parties; identity relationships are used by the party for providing (usually certified) identity statements about itself to other parties; the data track is used for keeping a record of all releases of data to other parties; and the profile data comprises data known about other parties. Each party needs to store those kinds of data in order to be able to use our architecture for privacy-enhanced identity management.

The above mentioned kinds of data need a well-defined way of representation for their formal modeling. The current section discusses the different kinds of data held by each player in our system. In Section 9.3 we discuss how data can be modelled formally for representation, storage, communication (between parties and components), and reasoning purposes. We already anticipate here that the data held by a party are expressed in a formal data model based on formulae in a fragment of first-order logic (FOL). Intuitively, such a formula is able to express statements about identity data (attributes) related to parties as well as part of the metadata over those data. For all classes of data, this concept is used for the data representation. We show in detail how the data model is applied to represent the different classes of data in Section 9.4. We stress that every party in a system maintains the same kinds of data, but that there are differences in the purposes the data are used for depending on the kind of party. We note that a record of any of the kinds of data we present has a holder and a subject associated, with the holder being the party that stores (holds) the record in its data processing system and the subject being the party the record is about.

9.2.3.1   Identifier Relationships

Our architecture is, as mentioned already, targeted at interactions between parties where parties are not necessarily revealing their legal identities to each other. They rather use identifiers that do not comprise any other attribute semantics than being an identifier for the party in a specific interaction with another party. Such an identifier is used as a mutually-known reference to the subject this identifier refers to – some kind of reference is required to be able to refer to a party. A party can have many such identifiers, even multiple ones with a single other party, and control by itself whether it wants to link any of those or keep them unlinked in the view of the other party. We call such identifiers *subject identifiers*. A subject identifier can refer to the party itself as is the case for pseudonyms of a user with other parties, or it can refer to other parties which is, for example, the case when a service provider makes statements about data about one of its customers to another service provider while using different names for the customer for different parties it talks to or in different interactions. The concept of subject identifiers thus is a generalization of the concept of pseudonyms because a pseudonym is defined as being only an identifier of the party that uses the identifier [PH10]. In order to be able to implement the concept of subject identifiers, a party holds identifier

relationships as one specific kind of data. The party uses these identifier relationships to maintain the subject identifiers it has established and to address the subjects of those in interactions with other parties and make statements about the subjects. An identifier relationship can be established by a party towards one, a subset of the other parties, or all other parties in the system. Both the party and the recipient maintain a record for the identifier relationship, flagged appropriately. We refer the reader to the terminology paper of Pfitzmann and Hansen [PH10] on a discussion of pseudonyms and other concepts of privacy-enhanced identity management.

Example (Subject identifiers): For example, a user who is, within the system, uniquely identified with the name (identifier) jane1234 can use the random name user4567 in one interaction with one party and the independently-chosen random name user6789 in another interaction with the same or a different party. In both cases, the name refers to the same subject (user) jane1234 and acts solely as an identifier without conveying any attribute semantics. To continue the example, the party that knows user jane1234 under the name user6789 can make statements about her under a different identifier user1357 to yet another party.   □

All of this is modeled with the concept of identifier relationships. Attribute data can be associated with the name as will be shown later. The preferred use of a subject identifier is semanticless use, thus being like a random identifier to which attribute data can be associated. This is the use of a pseudonym that is well known in identity management [PH10]. For parties such as service providers who always use the same identifier with their communication partners, it can make sense that they expose a public identifier towards all parties in the system. This is the case for service providers on the Internet as they expose a publicly-known identity towards every party in the system, and thus they can also expose a public identifier to every party in the system.

9.2.3.2   Identity Relationships

One important functionality of our architecture is the privacy-enhanced release of certified data. This requires that a certifier $\mathcal{C}$ decides to vouch for data about subject $\mathcal{S}$ towards other parties. The decision and related information of a certifier vouching for data about $\mathcal{S}$ is expressed through an *identity relationship* established between parties $\mathcal{A}$ and $\mathcal{C}$, with an indication of the role of the party (certifier/certifying party or subject/certifiee/certified party). The prominent case in the setting of user-centric identity management is that $\mathcal{A}$ and $\mathcal{S}$ are the same party. An identity relationship is a protocol-agnostic representation of such a decision to vouch for identity data of a party by another party as well as a representation of the data being vouched for and metadata.

A party maintains an arbitrary number of identity relationships with an arbitrary number of certifiers. An identity relationship specifies the following:

the data about $\mathcal{S}$ which $\mathcal{C}$ has decided to vouch for towards other parties and metadata about those data, including information about the certifying party as well as about the validity of the certification. Identity relationships are used by $\mathcal{A}$ to communicate (certified) identity data about $\mathcal{S}$ to other parties, with or without the online involvement of $\mathcal{C}$, depending on the protocol. An identity relationship can be technically realized with a range of protocols with different properties.

Examples for identity relationships are the following: an electronic passport obtained by a user from the government of her home country; an electronic driver's license obtained by a user from the responsible administration; a user's electronic subscription to the premium edition of an on-line newspaper; an electronic degree in accounting that a user has obtained from an on-line course provider; an accreditation of good privacy practices (privacy seal) that has been awarded to a service provider; or an identity certificate of a service provider. Numerous more examples will come to the reader's mind when thinking of the future electronic society and "credentials" that will need to be available for an interoperation between parties.

Example (Electronic identity card): An informal example for the data expressed by an identity relationship representing an electronic identity card that the user Jane Doe has obtained from the Swiss government is presented next:

Subject of the identity relationship: jane1234. *Identity data for Jane Doe:* firstname: Jane; lastname: Doe; gender: female; birthdate: 1977-12-24; country of residence: Germany; idcard serial number: fq3854390976. *Metadata about the certifier:* unique identifier of certifier: http://switzerland.gov/idcardissuer. *Metadata on the validity of certification:* valid from: 2009-07-01; valid until: 2014-06-30.

The Swiss Government vouches for those data as the certifier and user Jane Doe as the holder of the identity relationship can use it to release parts of the contained data to other parties.   □

An identity relationship is used by its holder to release the data or parts of the data contained therein to other parties in certified form, that is, consistent with the data the certifier agrees to vouch for. A single data statement to be released can comprise data from different identity relationships. Our architecture is designed to allow for different protocols to be used for revealing data based on identity relationships to another party. The protocol we focus on in this work, because of its strong privacy properties and the possibility to extend with strong accountability properties without reducing privacy, is a private certificate protocol called *Identity Mixer* [CL01b, CL03, CL04] which allows for strong privacy properties when revealing data, e.g., by revealing parts of the data of identity relationships or proving predicates such as the less-than predicate on attributes instead of revealing their values, and particularly by the unlinkability of transactions at the protocol level. A protocol like the Identity Mixer private certificate system allows the holder of the identity relationship to release certified data based on it without involving the certifying party.

We conclude this introduction to identity relationships with the note that the holder and certifier of the identity relationship can be the same party in which case the relationship refers to a *self-issued* or *self-certified* identity. Such an identity is purely a statement by the holder without any endorsement through a third party. Still, it has relevance in our architecture for use cases when such party-declared attributes are required. Particularly, this resembles the functionality of today's automated form-filling embedded within the data exchange functions of our architecture. We explain in detail in Section 9.4.2 how the data of an identity relationship are expressed in our data model.

9.2.3.3   Data Track

Data tracking refers to the idea of a party keeping track of which parties have obtained what data from this party, regardless of whether the party itself is the subject of the data.[3] The subject of the data can be either the party itself or any other party the party is releasing data about. This allows the party to later make an assessment of to which parties certain data items have been divulged to. For a user, this is particularly interesting in relation to the *access to data*, a right users are given by the European Data Protection Directive [Eur95]. For a service provider, this mechanism ensures that every data release of customer data to a third party can be tracked and, if required, used to support a user in their access to data and also to potentially keep user data up to date.

The data track of a party consists of data track entries. Each entry comprises the following: the data the other party has obtained, expressed through a formula in our data model, and metadata over the data, such as information about the recipient, the agreed data handling policy for the data, or the date of release.

Example (Data track entry):   An informal example for a data track entry based on the identity relationship further above is the following. The record models only the subject identifier, the gender, a statement on the birthdate that allows for inferring an age greater than or equal to 18 years taking the release date of the data of 2010-05-30 into consideration, and the country of residence being Germany as well as metadata. *Released identity data:* subject identifier: user4567; gender: female; birthdate: before 1992-05-30; country of residence: Germany. *Metadata about the certifier:* unique certifier identifier: http://switzerland.gov/idcardissuer. *Metadata on the validity of certification:* valid from: before 2010-05-30; valid until: after 2010-05-30. *Metadata on the release:* released on: 2010-05-30: data recipient: service3915.

---

[3] Recent work has put forward the idea of tracking data releases done by other parties as is relevant in situations such as social networks. We do not consider this, but note that our model of the data track can also represent such data, if appropriate protocols are implemented to communicate the updates of the data track.

In addition to the data given in the example, the data handling policy mandating how the recipient needs to handle the data is also included in the data track entry for reasons of accountability and for inquiring on the enforcement state of it.   □

If based on identity or identifier relationships, the statement expressing the released identity information can combine parts of the statements expressed by formulae of multiple identity relationships, if the underlying protocol for data release supports such a combination. Using a different example than the above, the first name, last name, and nationality attributes can be taken from an identity relationship representing the party's electronic identity card, and the statement that the party is allowed to drive a heavy truck with a hanger can be taken from the identity relationship representing the party's electronic version of her driver's license.

The data in the data track together with the metadata allow a party to gain a complete transaction overview in terms of data releases to other parties.[4] We give the most prominent uses of the data track next for different types of parties in the system: For a user, the data track can be utilized for the following purposes: manual inspection of releases by a user; automated update of changed attributes; and exercising all functions of access to data. A service provider can use the data tracking information for multiple purposes as well: making it available to the user who is the subject of the released data, thereby allowing the user to make use of their access right to the data even at third-party recipients of the data; carrying out user requests for the access to data held by third-parties on behalf of the user. Either a user or a service provider can use their data track information for the following: querying the enforcement state of the agreed data handling policy; access to the data; and as evidence serving accountability purposes, e.g., for the case of disputes. Other kinds of parties may have different uses of a data track. Considering the proposed uses of the data track, we conclude that it is an integral source of information for supporting a user in her identity management.

9.2.3.4   Profile Data

Within an identity management system, each party needs to store data about other parties it interacts with or has relationships with. Such data about other parties the holding party has obtained is denoted *profile data* in this work where our notion generalizes the usual notion of profile data.

The profile data stored by a party consists of profile data records (entries). Each such record is related to one party, the record's *subject*. Each entry

---

[4] Note that the identity management system of a party only captures data releases performed by the system, but not such that are performed "out of band" by the user herself, e.g., through Web forms. Our architecture can account for this through an extension that captures data releases through the Web browser and reports them to the identity management system.

comprises the following: data about the subject, represented as a formulae in our calculus; metadata on those data.

Example (Profile data entry): Continuing the series of informal examples from above, the release of the data by our example user Jane Doe results in a profile data entry as follows held by the data recipient known to the public as service3915: *Released identity data:* subject identifier: user4567; gender: female; birthdate: before 1992-05-30; country of residence: Germany. *Metadata about the certifier:* unique certifier identifier: http://switzerland.gov/idcardissuer. *Metadata on the validity of certification:* valid from: before 2010-05-30; valid until: after 2010-05-30. *Metadata on receiving the data:* received on: 2010-05-30.   □

The reasons for storing profiles about other parties are manyfold: For a user, her profile records comprise the identity data known about service providers, certifiers, and other users she interacts with. The reason for a user to keep those is on the one hand to enrich the data track information with service provider information such as privacy seals the provider has released such that the presentation of data track information to the user can be enriched with such information on whom data has been released to, as well as to maintain required information about other users and service providers, such as reputation statements obtained from third parties, that may simplify future interactions with those parties. For a service provider, the profile data comprises all identity data known about other parties, that is, for example, the customer profiles of its customers. Such profiles are congruent with the common notion of profiles. Profile records are one of the key resources of a service provider to be protected and properly handled in terms of user data protection as they may comprise personal data of users. The need for a profile exists as a service provider may need to store certain data about its customers in order to provide a service or retain certain data for legal reasons. In addition to profiles, a service provider also maintains data about other service providers it interacts with and certifiers.

It is important to note that data about other parties are stored in a profile related to the other party independent of the origin of the data: Data can have been provided by the subject itself, but also by third parties. The further is the case of a user releasing attribute data about itself, e.g., by registering their identity profile with a service provider, the latter is the case of a third party receiving (parts of) a user profile from the service provider for secondary use.

We stress that the profile data of all parties also comprises data about certifiers, that is, attribute information about parties issuing identity relationships to other parties. This allows for handling the certifier information within the certification metadata of formulae equally to any other data. The advantages of this uniform treatment of parties will become clear in Section 9.3 on the data model when we explain the mechanics and applications of the data model.

9.2.4   Components

We next present a high-level overview of the main components of the architecture. A selected set of those components or the mechanisms they implement are discussed in this chapter. Further details on those and the remaining components of the PRIME architecture will be presented in chapters of the book following this chapter on the architecture.

9.2.4.1   Identity Management

The identity management component implements functionality for assisting people in their identity management decisions and processes. Its main functionality comprises the following: assisting a user in the selection of her partial identity to use in an interaction with a service provider or other user; providing management functionalities for partial identities; automating the process of updating attribute data of the user at (a subset of) the service providers it has previously been released to as well as transitive data recipients; accessing data for the purpose of inspecting them, rectifying the data or requesting blocking or deletion. In performing its tasks, the component is a component that orchestrates the intra-party message flows at the party and performs inter-party message flows.

The component particularly relies on the negotiation component for ensuring mutual attribute-based authentication for ongoing interactions and the enforcement component for protecting access to the party's data and other resources. The component has an extensible architecture such that new functionality can be added by adding modules to it.

9.2.4.2   Authorization

This component implements a stateless authorization policy evaluation engine. The engine can evaluate authorization requests to resources of the party. Relevant state information is passed to the component with each invocation. In contrast to traditional authorization architectures where the authorization engine answers a request with an answer from the set $\{grant, deny\}$, our component allows for a third possibility. This third answer is a request for data and is issued if neither a *grant* nor a *deny* response can be returned given the current request, that is, if the request cannot be decided on without the requested data being provided. The data request determines the subsequent authentication steps of the other party based on which the original request to the authorization component may be authorized at a later point. A data request output by a service provider a user requests a service from is the standard case as users start interactions anonymously and service providers typically need some information about a party before releasing services to it.

The architecture of allowing this third kind of answer has been originally put forth by Bonatti and Samarati [BS02b] and consolidated and extended

with data handling aspects by Ardagna et al. [ACDS08]. A detailed architecture has been put forth and implemented in an effort during the PRIME project [PRIa]. We build on the same model for a variety of advantages compared to traditional authorization models: the requestee (e.g., service provider) does not need to communicate its full policy upfront and can obfuscate parts of the policy that it needs to keep private; only relevant policies for the current resource under access are communicated; the requester (e.g., user) can obtain precise information on exactly the parts of the policy she still needs to fulfill and provide precisely this information; the capabilities of generating such data requests is the basic feature on which we have built a powerful yet practical privacy-enhanced negotiation protocol for mutual request and exchange of data.

9.2.4.3   Negotiation

The *negotiation* component provides the functionality for a mutual request and exchange of data between two parties as well as an optional agreement on data handling policies for the data to be exchanged. In our architecture, we build on top of the functionality of the authorization component in order to realize the negotiation functionality. That is, the negotiation component makes invocations at the authorization component in order to obtain information on how to proceed with the negotiation protocol. This architectural idea has originated early in the process of building the PRIME architecture and allows one to construct a practical negotiation protocol from our authorization component.

   The negotiation component is used whenever a party starts an interaction with another party and the other party requires some data to be released in order to proceed. This triggers an instance of the negotiation protocol, or a negotiation in short. A negotiation can be seen as a mutual authentication based on (certified) data including an agreement of data handling policies to be applied to the data. Each negotiation is determined by the initially-requested resource as well as the authorization policies of the involved parties. A negotiation proceeds, once triggered, with the mutual request and release of data.

9.2.4.4   Data Exchange

The *data exchange* component implements protocols for the privacy-enhancing exchange of data between parties. This includes the establishment of identifiers for parties, the establishment of identity relationships, that is, the decision of a certifier to vouch for specified data of a subject, the release of data, that is, the use of the identity relationship, the revocation of identity relationships, as well as a protocol for escrow-like identity handling.

   The component shields the complexity of the implemented protocols from the other parts of the architecture, while having a powerful interface exposed,

based on our data model. Only the interface needs to be known to other
components that intend to use the services of the data release component while
protocol-specific aspects are hidden from the other parts of the architecture.
Regarding complexity of implementation of data exchange protocols, we want
to note that an implementation of a private certificate system such as the
Identity Mixer system results in one of the most complex subsystems of the
architecture.

9.2.4.5   Logic Reasoner

The Logic Reasoner component implements functionality for making deriva-
tions over the logic our data model is based on. This can, for example, be
used for deciding on whether a set of formulae allow one to derive another
formula, useful for the computation of how a request for data can be fulfilled
with data the party holds. Another example is deriving the resources a target
of a policy rule expresses. A further example is checking whether a list of
formulae satisfies the part of an authorization policy rule expressing its data
requirements.

9.2.4.6   Data Repository

The data repository holds all the data of the party. This comprises all kinds
of data introduced in Section 9.2.3. The component can be accessed through
simple queries and return sets of results similar an SQL database, though
specific to identity data in a privacy-enhanced setting.

9.2.4.7   Policy Repository

The policy repository holds all authorization policies, data handling policies,
and negotiation policies of the party. The component can be accessed through
queries and returns sets of results.

9.2.4.8   Console

The console is the user interface component of the architecture. It imple-
ments the user interface concepts for the identity management functionality
of the architecture. This particularly comprises the following: selection of the
data to release for answering a request of another party; customization of
the data handling policies for data to be released; interactive requests of (as-
surance) data from the other party; displaying information required to give
informed consent for a data release, particularly including policies and in-
formation about the other party; giving informed consent for a data release;
administration of ones's policies and preferences; access to data.

9.2.4.9   Other Components

In addition to the components that have been outlined above, further components are included in our architecture, but not discussed in detail in the architecture chapter of this book, e.g., components for secure anonymous communication (see Chapter 18), policy management (definition and maintainance of policies), assurance and trust assessment, or life-cycle data management. See chapters following in this part of the book and also previous versions of the architecture document for details on those.

9.3   Data Model

This section discusses the modeling of data in our architecture. As mentioned in Section 9.2.3, there are multiple kinds of data to be modeled, such as the identifiers a party uses to address itself or other parties, a party's releasable identity data in the form of identity relationships, profile data held by a party about other parties, or data released by a party to other parties. Furthermore, the parts of authorization policies that define the (attribute) data requirements on the requesting party and the properties of the accessed object, as well as data requests and data statements communicated between interacting parties must be captured by our model.

In a practical system, different parties need to interoperate with each other, that is, need to understand identity requests and statements being made. This requires a common and mutually-understandable formal language being used for the interaction between different parties as well as relevant parts of policies that other parties need to act upon. Such a language has the same meaning—or semantics—for all parties in the system. Thus, such a language is a prerequisite for achieving interoperability for identity management between different parties in a system. Within the scope of a single party's system, data need to be processed, for example, stored, retrieved, being used for authenticating an other party, or access being controlled, data handling policies being enforced, or reasoning being done on them. Many of the processing steps within a party require an understanding of the meaning of the data. Using the same representation for data avoids performing mappings between different representations at different places, each such mapping requiring a formal definition and an implementation, thereby greatly simplifying the architecture. For those reasons, we have decided to represent data through our data model also within the scope of a party.

The result of our efforts on data representation is a unified model being applicable for both communication of requests and statements between parties as well as processing within a party. In terms of expressiveness, the language is able to model a wide range of statements about entities (parties and objects) and at the same time to allow for the parties' privacy to be protected. The design has been strongly governed by the concept of data minimization,

that is, the concept of the minimum possible amount of data as requested by the other party being revealed in a transaction. Particularly, this includes the expression of disjunctions of statements as well as predicates used to reduce the amount of information being revealed about attributes, instead of always revealing the attribute values. We note here that data minimization is a concept that, if it is to be realized, does not require only support at the technical level, which we are discussing in this work, but also at the level of business processes that need to be defined accordingly to work with the minimum amount of data possible. Also legal considerations come into play when discussing data minimization, e.g., what happens in the case of a dispute if a user is anonymous, or whether anonymous interactions are legal. The non-technical issues are equally important to the technical ones and have been treated in other parts of this book. This chapter focuses on technical aspects of privacy-enhanced identity management.

In this section, we present the formalism for modeling the different kinds of data and data requests we need in our architecture. The resulting data model specifies the syntax and semantics of representations of identity data. Based on the data model, we give insight into certain kinds of processing of data, e.g., how a satisfying data statement based on the identity relationships of a party can be found for a data request. In this book chapter we do not give the formal semantics, but leave this for future work on the data model.

The core of our model is a language based on first-order logic which is the main subject of discussion of this section. We start our discussion with the basic concepts underlying our language and then extend it with further, more advanced, concepts. Our presentation is guided by examples for illustrating the introduced language concepts to the reader in an intuitive form. Our contribution is a language that allows one to express identity information about entities in a general way and that is particularly suitable for the use with private certificate systems, today's most privacy-protecting mechanism for authenticating users to other parties, as data exchange technology. As already mentioned, we stress again that a concrete data model as we propose is a necessary precondition for a deployment of such private certificate systems in practice because an expressive and machine-processable representation of the identity data with clear semantics is required for integration with authorization and negotiation frameworks.

9.3.1   Identity

A foundational concept in our formalism of representing identity data is the *identity*. An identity is essentially a named group (set) of attributes with their values. Precisely, it is a named set of tuples comprising an attribute name, an operator, and an attribute value each. Below, we give an example identity c1234 comprising the attributes *firstname*, *lastname*, and *income*.

$$c1234 = \{(\mathit{firstname}, \mathit{Eq}, \mathit{Jane}), (\mathit{lastname}, \mathit{Eq}, \mathit{Doe}), (\mathit{income}, \mathit{Geq}, 3000)\}$$

Terms referring to identities are the basic building blocks of our data representation language. In the language, an identity can be referred to through an individual constant or a variable, both being *terms* of our logic-based language, e.g., the constant term c1234 of the example below. As a shorthand notation, we introduce the ".".-notation for qualifying the identity like a record type for referring to its attributes. The following is an example for referring to the attribute *firstname* of identity c1234 and saying that it is equal to the individual constant Jane. We do not go into the details of the formal semantics of our language in this book.

$$Eq(\text{c1234}.firstname, \text{Jane})$$

An identity may characterize a party in terms of the party's attributes of its *civil identity*, such as its name, address, date and place of birth etc., other *assigned attributes*, such as the name and grade of a course a user has completed on-line, or *assign rights* to the party, e.g., specify the rights of the party for accessing an on-line resource such as for a subscription to an on-line newspaper or movie store. Although, all those cases are different in terms of what parts of the identity of a party are concerned, there is, from a technical perspective, no need to handle those cases of the identity of a party (in terms of attributes) in the strict sense and rights assigned to the party, differently. Thus we subsume all of those into the concept of identity. This gives rise to a wide meaning of the term *identity* in the data model. In the sequel, the use of the term identity should usually be clear from the context it is used in, otherwise, we explicitly clarify it. Other semantically meaningful names to refer to the identity concept are *attribute group* or *attribute set*. We chose the term *identity* for its genericity as well as the fact that it is related to the concept of *partial identity* as it is well known in privacy-enhancing identity management research.[PH10] A partial identity is the part of a party's complete set of attributes it holds about itself that it exposes to another party in an interaction or a set of related (linked) interactions, whereby an identity in our meaning captures a part of a party's attribute information, but not in the context of its complete attribute information being revealed to another party.

An identity is typically used as a conceptual *grouping* of attributes, as it often occurs in real life, e.g., government-issued credentials such as passports, driver's licenses, or residence permits group attributes relevant in the context of each of those. This grouping provides additional semantics to the contained attributes of the identity, by stating that they belong together. For example, both an account balance and a currency attribute for a bank statement need to be grouped together, otherwise they will not have the intended meaning of denoting the account balance in its associated currency. Further below, we introduce the *type* of an identity modeled as an attribute which is one means of providing further meaning to an identity and its attributes.

From an identity management perspective, the concept of associating attribute information with identities and identities with parties—in particular

people—has the big advantage that people can have and use different attribute values for the same attribute as is commonly done in today's Web interactions by users using pseudonyms and picking different names for different accounts, for example, the various nicknames people on the Web can have in different contexts. This idea is foundational to privacy-enhanced identity management in general, see [PH10] for a detailed account.

An identity cannot change over time within a concrete system. Changes to an identity are implemented by establishing a new identity with the changed information and rendering the to-be-changed identity obsolete through attached metadata. This ensures that basic properties of the underlying logic are accounted for.

In the remainder of this section, we will introduce a set of reserved or predefined attributes of identities that have predefined meaning important for the purposes of identity management. Any other attributes but those can be freely defined.

9.3.2   Constants

Another basic building block of our language are constants. A constant is a value from a value domain, depending on the type of the constant. The language supports in its basic variant the types integer, date, and string. The type integer comprises all integers with a total order defined over them. Date is technically similar and provides essentially "syntactic sugar" over the integers for easier use of the language. The type string comprises all strings from a suitable alphabet. This basic set of types can be extended with additional types and predicates the signatures of which comprise arguments of those types.

The constants are elements of a typed universe and in our language they are referred to by terms encoding the constants, so-called self-referential terms. For example, the integer constant 10 of the universe is referred to by the term 10 in the language. In other words, a constant term is interpreted with itself in an interpretation of a sentence (formula) in our data model.

9.3.3   Formulae in First-Order Logic

The basic entity for representing identity data is a *formula* in a fragment of *first-order logic*. Such a formula can be used for representing data at different places, such as in an identity relationship, in a data track, in profile data, in a data request, in a data statement made to a party, or a data requirements specification in an access control policy. A formula thereby expresses two kinds of things: *identity data* related to one or more subjects and related *metadata*. The identity data comprises information on attributes related to the subjects while the metadata comprises information on the certifiers of the identity data, the temporal validity of certification, and possibly other metadata. All of this forms a unit, the formula in our data model.

Our language allows for specifying predicates over identities and their attributes as defined below and connecting such predicates with the standard ∧ and ∨ connectives of first-order logic to build up comprehensive formulae making (data-minimizing) statements about parties. Particularly the possibility of the ∨ connective greatly improves the model in terms of data minimization functionality compared to the standard name-value pairs of today. Furthermore, the language is able to express parties' data together with certification metadata for the data. This is useful in terms of integrating both data and trust aspects of the data in a single model and allowing for policy decisions based on both.

In the remainder of this section we introduce the fragment of first-order logic used in our work for modeling data. It has sufficient expressiveness to satisfy, from a data model perspective, many use cases we have in mind for user-centric privacy-enhancing identity management. The fragment will be introduced in a step-by-step manner, with explanation of the underlying concepts and examples for illustration.

9.3.4 Predicates

The example further above has already made use of the concept of predicates as is standard in first-order logic: Our formulae are built up from predicates to express relations between attributes of identities, constants, and other objects introduced further below. The predicate *Eq*(...) with its two arguments above, for example, expresses that the attribute *firstname* is equal to the constant Jane.

In our language we support a set of predefined predicates, depending on the data types of the arguments. For integer and date arguments, we allow the predicates *Eq*, *Neq*, *Lt*, *Leq*, *Gt*, and *Geq* which are the standard relational operators on totally-ordered sets with their standard meaning: Equal, not equal, less than, less than or equal, greater than, and greater than or equal. Negations of each of those can be expressed as is standard, with a corresponding predicate: The negation of less than can be expressed through greater than or equal, for example. For strings we define the predicates *Eq* and *Neq*. This is a restricted set of predicates, the choice of which has been governed by the requirements for privacy-enhancing identity management based on data minimization as well as what can be efficiently implemented in practice by private certificate systems for exchanging data. The language and its semantics can be extended with further predicates if this is required in the future.

9.3.5 Connectives

Two or more predicates are connected to a formula by using the standard ∧ and ∨ connectives of first-order logic. We allow parantheses in the standard

way to be used for setting precedences that deviate from the built-in precedences of the language. Standard precedences as known from first-order logic languages apply: conjunction binds more strongly than disjunction.[5]

Example (Predicates): The following is an example of a formula comprising two predicates over an identity connected with the conjunction connector $\wedge$.

$$Eq(\text{c1234}.\mathit{firstname}, \text{Jane}) \wedge Eq(\text{c1234}.\mathit{lastname}, \text{Doe})$$

$\square$

9.3.6   Subject

Our data model can associate an identity with a *subject*, where the subject is the entity or party which the data represented through the identity is about. The association is done through a subject term assigned to the *subject* attribute of the identity. We establish the convention that this attribute be available for each identity.

$$\phi = \ldots\, Eq(\text{c}.\mathit{subject}, \text{user4567}) \ldots$$

The party that is the specified subject of an identity remains the same for the identity at all times, though the party can be referred to via different terms in different references to the subject, e.g., in different formulae, as is possible in first-order logic. Technically, this means that in an *interpretation* of any formula talking about the identity, the subject term of the identity always maps to the same party being the subject.

Considering the possibility of multiple constant terms referring to the same subject in an interpretation, the *subject* attribute is different to other attributes in this respect as one may use different terms for the subject term to refer to the same identity and thus the term is not self-referential as is true for other attributes. Note that exactly for this reason and also the reason that it would not be workable from a conceptual point of view, we do not allow that the subject attribute be related with an attribute of a conditionally-released identity as introduced further below.

When a party derives a new formula from an existing formula, e.g., an identity statement ϕ' from a formula ϕ of an identity relationship, it must follow the following constraints on the renaming of the subject in ϕ': The term for the subject in ϕ' must be an element of the set of terms representing the names (pseudonyms) for the party established between the party and the intended data recipient of ϕ'. Note that depending on the technology used to realize such names and thus also the binding of identities to parties, pseudonyms can be implemented through cryptographic means, e.g., through

---

[5] For a language to avoid parantheses, one can use Reverse Polish Notation (RPN) for expressing formulae and precedences of connectives without a need for parantheses.

the protocols of private certificate systems such as [CL03, CL04], or they can be based on trust in the party, e.g., a service provider releasing data about one of its customers to another service provider and making a claim about the subject identifier. In the further case of private certificate systems, enforcement of the use of only correct identifiers is done by cryptographic means, while in the latter case trust in the service provider is required that it is using the correct subject identifier.

In the below example ϕ_1, derived from the formula in the above example, the identity has been renamed to c_1 and the subject to the term user6257. Example ϕ_2 is another (different) use of the same formula with different terms referring to the identity and subject. This reflects the typical use of a formula ϕ from which new formulae are derived, using different terms for the subject and identity, to release the derived formulae to interaction partners using a private certificate system and thereby not introducing linkability.

$$\phi_1 = Eq(c_1.firstname, \mathsf{Jane}) \wedge Eq(c_1.subject, \mathsf{user6257})\ldots$$
$$\phi_2 = Eq(c_2.firstname, \mathsf{Jane}), \wedge Eq(c_2.subject, \mathsf{user8634})\ldots$$

9.3.7   Identifier Objects

In addition to the concept of identities we use the concept of identifier objects to model identifiers established between a party and other parties, about potentially other parties. An identifier object is established through the protocol for establishing an identifier relationship of Section 9.6 as the main part of the created relationship. Once this has been done, the relationship and its identifier object can be used to prove holdership of the object to any of the parties it has been established with. The special, yet most important case that the subject and holder of an identifier object and the corresponding identifier relationship are the same party, equals the concept of a *pseudonym*.

An identifier object is syntactically represented in a similar way as an identity, though it is different in that identifier objects can only be released to, by proving holdership, the parties they have been established with and that they have an exhaustively-specified set of allowed attributes used to express their properties, while identities can be used with any other party and can have arbitrary attributes associated with them in addition to the predefined attributes. Due to these differences, identifier objects and identities cannot be cleanly modeled by a single concept, though, their modeling is closely related. An example of an identifier object is given next:

$$Eq(\mathsf{p}.subject, \mathsf{user4567}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567})$$

As the example shows, the identifier object p has an attribute *subject* that comprises a term for the subject party of the identifier object which may be referred to by different terms for different uses of the identifier object. The attribute *subjectId* is a constant term used as the identifier of this identifier

object and must not be renamed throughout different uses of it. Also, p itself is not renamed when using it because the purpose of an identifier object is to establish linkability between the interaction of its establishment and all of its use interactions. Any use of a subject attribute of an identity in a formula to be released is constrained in that only terms must be used for the *subject* attribute that are used for the *subject* attribute of identifier objects established with the party the formula is intended to be released to. This reflects the natural property of an identifier object representing a shared identifier for a party used by a party with other parties.

A subject identifier is not only relevant for the use case where a user communicates with a service provider and uses the subject identifier in this communication to refer to itself. In a scenario where two service providers interact to exchange customer data, the concept of subject identifiers equally applies. Though, the trust model is different in that in the former case, cryptographic protocols are employed to enforce the correctness of subject identifiers while in the latter case, the service providers trust each other in using correct identifiers. As the concepts are equal—a party talks about another party, possibly itself, towards another party—we use the same concept to model it, while protocols with considerably different properties can be used for implementing the concept of identifiers of parties.

Domain Identifier Objects

A domain identifier object is a special kind of identifier object with the meaning that a party may only obtain a single one with one other party comprising the same domain string. The domain string can be freely specified by the party the identifier is established with and typically delimits different scopes of this party where it is a requirement that other parties are known under a unique identifier. For example, in an e-learning service, the service provider may require that users can register only under one identifier per course and thus require domain identifier to be used:

$$Eq(\mathsf{p}.subject, \mathsf{user4567}) \wedge Eq(\mathsf{p}.subject, \mathsf{user4567}) \wedge$$
$$Eq(\mathsf{p}.domain, \mathsf{Elearning\_course\_Finance})$$

The use of such identifier objects allows a service provider to restrict users to a single registration for a service. A cryptographic pseudonym system such as Identity Mixer allows for enforcing the uniqueness of such identifier objects. When proving holdership of a domain identifier object, its domain needs to be always revealed, otherwise the underlying cryptographic protocol must terminate with failure as the purpose of the domain restriction would be defeated otherwise. In the special, and most relevant, case of the subject and holder being the same party, the concept is equal to the concept of *domain pseudonyms*.

9.3.8 Certification Metadata

Certification metadata are data associated with an identity and describe aspects related to the certification of the identity. This includes specification of the certifier, e.g., by identifying it through a single attribute, or specifying it through a combination of attributes, the temporal validity of certification, and protocol-related parameters. Those identity metadata are required in order to allow a recipient of a data statement based on the identity to make their trust decisions on the data as well as to express the certifiation requirements when requesting data (authentication). Without such certification metadata being available on identity data, it is often not possible for a recipient of the identity data to make a decision on whether to trust the data for the purpose at hand. For this reason, one may claim that those metadata are equally important for making a policy-based access control decision than the identity data themselves they are associated with. We next show how certification metadata are expressed in our data model.

Example (Certification metadata): The example expresses that party c is the certifier of dl and that c is as well the subject of the identity cid. Via this identity cid, statements about the certifier c can be made, here only the statement is made that its the value of its attribute *uniqueid* equals German_Government.

$$\ldots \wedge \textit{Eq}(\mathsf{dl}.\textit{certifier}, \mathsf{c}) \wedge \textit{Eq}(\mathsf{cid}.\textit{subject}, \mathsf{c}) \wedge$$
$$\textit{Eq}(\mathsf{cid}.\textit{uniqueid}, \mathsf{German\_Government}) \wedge \ldots \qquad \square$$

As can be seen from the example, we embed the certification metadata directly into an identity statement, that is, into the formula for expressing the data. Technically, we associate an identity with the certifier specified as the identity of the certifier or the "main" identity we are talking about in the formula. The identity of the certifier can be described through predicates as they are used for representing any other data in our language. That is, the statements about the certifier form a sub-formula expressed over the certifier's identity. The example introduces the term cid for referring to the identity of the certifier c and specifies the attribute uniqueid to be equal to German_Government, assuming that the attribute can uniquely identify parties using a string. Clearly, more complex sub-formulae as in the example can be used to specify the certifier's identity, including the use of disjunctions. This makes the language expressive in terms of referring to any of a set of certifiers with the properties specified through the identity. This is a useful property for expressing certification requirements in an access control policy. Using this approach is also very dynamic as a policy author is free to refer to any attributes of certifier identities in the specification of the certification requirements.

We argue that the idea of specifying a certifier by once again reverting to the concept of identities is the natural choice as there is no strong reason to introduce an additional concept for data modeling for parties acting in the role

of a certifier and thereby complicating the language. Particularly, identities are issued to certifiers in a way that is conceptually the same as identities being issued to users. Thus, each party is described via identities in our model regardless of what kind of party it is, that is, under which combination of roles of user, service provider, certifier, or other roles it acts. There is no conceptual or modeling difference between identities with their subject being a certifier or identities with the subject being any other kind of party, e.g, a user.

Technically, an identity of a party not being a certifier comprises an attribute named *certifier* which can be related to a certifier's identity that describes the certifier of the identity. For the certifier identity, the same concepts apply as for any identity: it comprises a set of attributes and predicates can be expressed over those in order to specify it. The exception is the case of the party itself being the certifier for the identity; this case is handled by not specifying the certifier or by associating it with the subject identifier of the party. In the latter case, the subject identifier is renamed as usual, when running a prove protocol based on private certificates.

Delegation and Anonymous or Pseudonymous Certifiers

From a privacy standpoint, the constant term c referring to the certifier can be renamed to a fresh identifier in different formulae referring to the same party or always be the same identifier, following the usual rules for renaming. The latter approach of repeatedly using the same term can be used for most attribute exchange protocols, such as standard private certificate systems or traditional certificate systems as the certifiers are always identified parties with a unique public identifier. When considering the case of applying advanced technology, e.g., hierarchical credential systems [CHK+06] where a party gives a private certificate to another party in a delegation relationship, the party is not necessarily identifiable, but rather known by its attributes. A renaming of the terms for the party's identity and the party can be performed to allow for unlinkability of transactions in this setting and the party is then only specified through attributes. This advanced technology can be used for realizing delegation without the delegating party being identified. A different use case, mostly found in collaborative scenarios, is users who may become certifiers and issue identities to other users, while being known on the basis of their attributes rather than unique identifiers. Such use cases can be expressed easily in our data model by specifying a certifier through its attributes.

9.3.9   Conditional Release

The concept of *conditional release*[BCL04] makes it possible that a party releases identity information in a way such that only once a predefined conditional release condition is met, a previously-determined third party $\mathcal{T}$—the conditional data recipient—can obtain the conditionally-released identity information. Using cryptographic mechanisms, this concept can be realized in a

strong trust model [BCL04]. This may involve the conditional data recipient in the release interaction. In our formal language, conditional release is specified using *conditionally-released identities*. A conditionally-released identity is much like any other identity: statements can be made about the attributes of the identity through predicates in a formula, e.g., by relating them to other attributes. Like for identities, the attribute values of conditionally-released identities are not obtained by the data recipient during the data release interaction – it only learns the predicates expressed on it. We give an example fragment of a formula to show the use of conditionally-released identities within a formula to be released:

$Eq(\mathsf{c}.firstname, \mathsf{Jane}) \wedge Eq(\mathsf{c}.certifier, \mathsf{u}) \wedge \ldots \wedge$

$Eq(\mathsf{e}.serialnumber, \mathsf{c}.serialnumber) \wedge Eq(\mathsf{e}.condition, \mathsf{Misuse\_of\_service}) \wedge$

$Eq(\mathsf{e}.conditionalRecipient, \mathsf{t}) \wedge Eq(\mathsf{tid}.subject, \mathsf{t}) \wedge$

$Eq(\mathsf{tid}.party, \mathsf{Swiss\_revocation\_services})$

The example shows how the attribute *serialnumber* of conditionally-released identity e is specified to be equal to the attribute *serialnumber* of the identity c without revealing the latter. Furthermore, it shows how the conditional release condition is modeled as an attribute of the identity as well as the intended recipient being specified through another identity, as is done with other identities. The recipient is thereby expressed through the dedicated attribute *conditionalRecipient* in the identity e.

The choice of again using the identity concept for expressing the conditional recipient of the conditionally-released identity is motivated as follows: First, it conceptually fits the idea of using identities to specify attribute statements about parties, and thus is integrated into the model and derivations over it naturally by simply using the same language elements. Second, it allows for flexibly specifying a set of parties as possible recipients through appropriate specification of the predicates, which is particularly useful in a policy for giving choice of one of multiple data recipient parties to the data releasing party in a data request.

Conditionally-released identities are used along other identities in data release protocols, usually for establishing revocable anonymity through escrowed identity information that can identify the party once being de-escrowed. An actual revocation, the de-escrow, then requires additional protocol flows, depending on the exact scheme being used. Typically, the conditionally-released identity information for realizing anonymity revocation is the identifier of the party it had with the certifier at the time of creation of an identity relationship of another identity the conditionally-released identity relates to. We use the predefined attribute *subjectIdWithCertifier* which precisely models the pseudonym of the party with the certifier of the other identity. Details on this are given below. More generally, any technically-feasible and suitable combination of attributes can be conditionally released by a party (as required by

a data request), possibly to multiple conditional data recipients in a single transaction.

Formulae containing references to conditionally-released identities are applicable for making data statements to other parties, expressing the requests hereto, and storing those formulae in the data track and profile data. Though, such formulae are not applicable for modeling data in identity relationships.

9.3.10   Anonymity Revocation

In systems where users can be pseudonymous or anonymous in an interaction, legal regulations or other interests of parties may require that anonymity or pseudonymity can be revoked under well-specified circumstances. We introduce a reserved attribute *subjectIdWithCertifier* for identities that is set during creation of an identity relationship to the subject identifier the party has with the certifier at the time of the creation of the identity relationship; this attribute gets assigned the actual value of the subject identifier of the party with the certifier and is modeled as a self-referential constant, that is, it is interpreted with itself. It is crucial that it is enforced by the implementation of the system that the value of the attribute is the subject identifier the subject of the identity has with the certifier in the session of the establishment of the identity relationship. This approach of introducing a new attribute for modeling the subject identifier under which the subject was known when the identity relationship was created avoids to revert to more powerful logic for expressing this meaning through referring to the *subject* attribute. For realizing revocability of a specific transaction, the *subjectIdWithCertifier* attribute of an identity referred to in the data formula to be released is conditionally released to the data recipient, and can be obtained only by the specified conditional data recipient (trustee) once the associated conditional release condition gets fulfilled. The actual revocation of the anonymity, that is, obtaining the subject identifier, can be carried out by the trustee if asked so by the data recipient or a third party and after verifying that the condition holds. With this pseudonymous identifier, it is, depending on the setup, possible to obtain the identity of the party from the certifier of the identity. Conditional release has been introduced by Bangerter et al. [BCL04].

We note that the subject of an identity is not necessarily the party using the identity, e.g., in delegation use cases the subject is different from the delegatee. If one intends, in such cases, to allow for anonymity revocation of the anonymity of the delegatee (the acting party) as well, the policy must be phrased accordingly to refer to an identity the acting party is subject of, and not only holder.

9.3.11   Typing

We next explain the typing scheme underlying our language, comprising typing of the terms of the language through a typed logic as well as an additional typing mechanism for associating types with identities.

9.3.11.1 Typing through Typed Logic

All terms of the language are typed through using a typed first-order logic as a foundation. Such a typed logic extends plain first-order logic by associating types with all terms of the language. This is done outside of the language. Concretely, we associate data types such as integer or string to each attribute, the type identity to identities, the type identifier to identifier objects, the type cridentity to conditionally-released identities, and the type opaqueidentity to opaque identities. Proper typing according to this type system is a prerequisite to well-formedness of formulae in our language. We use the usual ::-notation for associating a type with objects of the language: cid :: identity. Types are stored and communicated with formulae – we usually do not mention the types explicitly when discussing the processing or storage of a formula for reasons of notational simplicity.

9.3.11.2 Typing of Identities

We allow that identities may, but need not, have an *identity type*—in addition to the one assigned in the typed logic—associated with them. This identity type defines the identity in terms of the ontology types of its attributes, the data types of the attributes, and technical features necessary to execute protocols associated with the identity. The type is technically realized as an attribute. An identity of an identity relationship usually has such a type, an identity (variable), e.g., in a data request, may not have one associated for reasons of greater expressiveness. The reason for having this type optional is to respond to requirements of real-world identity management systems: A prominent use case in such systems is to request attributes without imposing restrictions on the type of the identity the attribute is expressed through. This is an important use case when a party requests values of attributes that need not be certified by a third party and where the type of the identity they are contained in does not matter. We allow for such requests to be expressed in a succinct way through the identity concept by not specifying the type.

Static Type System

An identity may have an associated *type*, represented as an attribute denoted *type*. This attribute is metadata specified for the identity. A type hierarchy is induced by relating all types (type identifiers) in a type hierarchy through rules in our logic. For the case of using single inheritance, this gives a tree as inheritance graph. The inheritance graph is specified through the types being its vertices and the directed edges (subtype, supertype) for each subtype relation, thus implementing an *is a* relation. The static type of an identity specifies the ontology types (attributes) the identity comprises, expressed as a set of ontology types, and is referred to by its type identifier which is a constant term of the language.

The requirement on the identities of a type is that each attribute type of an identity of a supertype must be contained in all identities of its direct subtype. Furthermore, each identity of a type contains the same attributes. This adheres to the standard meaning of inheritance, e.g., as used in object-oriented programming languages. The issuers of identities of those types need to adhere to the above semantics of the type system. It is crucial that all identities that a certifier issues for a type comprise the same set of attribute types. If this is violated, identities with different sets of attributes will not be usable for fulfilling certain policies, thus violating the expected system behaviour. Though, it would not have detrimental effects on security, but rather availability would be compromised.

For implementing the subtype hierarchy, we define a predicate $Type(\_,\_)$ with arguments *identity* and *type* to specify types of identities and predicates of the form $Subtype(\mathsf{T}, \mathsf{T}')$ for expressing the hierarchy. Based on the native type of an identity as specified through its *type* attribute, as well as on the type hierarchy, the identity takes on all types upwards the type hierarchy as well and can be used at places where one of the supertypes is required. Rule of the following form implement the type system in the deduction system of our logic:

$$\forall \mathsf{C}, \mathsf{T}, \mathsf{T}' : Eq(\mathsf{C}.type, \mathsf{T}) \wedge Subtype(\mathsf{T}, \mathsf{T}') \rightarrow Type(\mathsf{C}, \mathsf{T}')$$

As basic case for our model, we permit single inheritance as it is sufficiently powerful for the requirements we have in mind within PRIME and it is conceptually cleaner as well. We do currently not have any specific use case in mind that would require multiple inheritance of identity types.

It is important that we do not assign the concrete type to an identity through sorted logic, but rather assign each identity only the generic type identity through this. The approach of typed logic for this purpose would restrict us from certain uses of our language which allow for elegant and powerful ways of expressing policy and data formulae, because this would rule out the dynamic typing and its applications as introduced next.

We do currently not allow for subtyping of attributes in subtypes of identities, the reason for this being that we do not have requirements for this and thus avoid the extra overhead.

Dynamic Type System

In addition to the (static) type system explained above, our definition of the language allows for a *dynamic type system* to be used for reasoning in our logic. The dynamic type system allows for an identity specified through a formula ϕ being of a *dynamic subtype* of another identity specified through ψ merely by ϕ referring to at most the attributes ψ refers to in making statements about their respective identities. This is independent of the static types specified through the *type* attributes. The latter must not be specified in a

formula if dynamic typing is to be used as otherwise the *type*-attribute would immediately constrain typing to the static scheme and rule out dynamic typing.

See the following example for a more concrete scenario and the relevance of the concept of dynamic subtyping in practice for the matching of a data formula against a data request. Consider as an example a data request asking for an attribute of an identity without providing any requirements on the certifier and type of the identity. This reflects the common case of a self-stated (declared) attribute being provided as is used in almost all interactions in today's Web through form filling. The example request by a service provider contains $Geq(\mathsf{C}.salary, 3500)$ as the attribute request part while not making any requirements on the certifier or the type of C. A user can fulfill this part of the request by using *any* of its identity relationships with an identity comprising the attribute *salary* which needs to be greater than or equal to 3500.

9.3.11.3 Architectural Aspects

The type information for identities needs to be distributed to the parties in the system in order to allow them to utilize these types. That means that for each new identity type getting vouched for by a certifier, the type information needs to be communicated to interested "consumers" of this type, among those being both users establishing identity relationships as well as service providers accepting data statements based on this type. The security property that must hold is *integrity* of the types. A simple approach is to store the type information in a suitable format and let the certifier sign this storage format with their signing key. The signed message can be obtained by usual means by any party in the system and verified in terms of integrity and associated with the certifier. For a certifier issuing private certificates, the cryptographic key used for signature verification can be the same one as the cryptographic key used for verifying zero-knowledge proofs based on private certificates issued by this party. This approach thus binds types to public keys, not restricting a public key to a single type for reasons of generality.

The processing of identity statements referring to a type requires a party who encounters an identity type that is unknown to them to obtain the type description and verify its data integrity. The latter is crucial for security. Only having completed this successfully, further processing related to identities of this type may be performed.

From a scalability perspective it can be expected that a large part of transactions of a party will be based on a reasonable-size set of identity types. Those can be obtained and verified once and retained by the party locally in order to avoid overhead when using or verifying identity relationships. Similarly, a user needs to obtain and check the type descriptions of identities it uses (in her identity relationships) only once and can then cache them locally.

We note that it is crucial that those seemingly trivial architectural aspects are accounted for in a real-world system deployment as they are important for security as well as availability properties of the system.

9.3.11.4   Discussion

The combination of our orthogonal type systems results in a flexible overall type system, combining advantages of a statically-typed and an untyped language in terms of typing of identities. Particularly, it allows for fulfilling real-world requirements of allowing for stating a request for attributes without referring to the type of the related identity.

9.3.12   Automated Reasoning

Automated reasoning is a powerful tool for considering formally-modeled knowledge of certain aspects of a system in the formal model of the data. Automated reasoning concretely allows for deriving, based on inputs that are assumed to hold, e.g., data statements proven by another party, facts that hold as well. The reasoning is specified through deduction rules. In the reasoning process, a new valid formula is derived in each step from the currently valid formulae by the application of one rule. In our logic we build on the rules of *natural deduction*, as discussed in [HR04], as the deduction rules of our logic.[6]

The basis for reasoning is a *sequent*, a standard logic concept, that expresses that a formula ψ (or a list of formulae) can be derived from a list of formulae $\phi_1, \ldots, \phi_k$ through the proof theory of the logic. This is expressed notationally as follows:

$$\phi_1, \ldots, \phi_k \vdash \psi$$

This sequent is true if it holds that we can derive from the list of formulae on the left side of the $\vdash$ symbol the formula on the right side, that is, assuming all the formulae on the left side are true, the formula on the right side is true. The derivation is done through successive application of the derivation rules of natural deduction on the formulae on the left side and the so-far derived formulae. The sequent holds, if the final derivation step allows for deriving ψ.

Based on this concept of derivation inherent to first-order logic, we define derivation in the logic based on an additional input element, a so-called ontology. An ontology $\mathcal{O} = \langle o_1, o_2, \ldots, o_l \rangle$ is a fact and rule base expressed through a list of formulae $o_1, \ldots, o_l$. For the decision on the truth of a sequent, the ontology is considered being a part of the formulae on the left side of the sequent, as shown next. Reasoning is then done exactly as explained above.

---

[6] We use a restricted set of deduction rules and obtain a reasoning system that is sufficient for our purposes and avoids certain problems when using the full natural deduction.

$$\phi_1, \ldots, \phi_k, o_1, o_2, \ldots, o_l \vdash \psi$$

We alternatively express this as follows:

$$\phi_1, \ldots, \phi_k \vdash_{\mathcal{O}} \psi$$

We require that all free variables in all formulae be instantiated through an *environment* $\mathcal{E}$ before checking the validity of a sequent. An environment thereby is an assignment of free variables with values from the value domains of those variables. Practically, an environment will comprise the concrete attribute values from identities and identifiers the party holds, that is, it is a function from free variables to constants.

Note that whenever we use the informal terminology of "a list of formulae implying a formula", we mean that an accordingly-specified sequent holds in our logic. Also note that intuitively a sequent as above means, in terms of identity information, that the formulae on the left side comprise at least the identity information as the one on the right side.

The *derivation relation*, or informally referred to in this work also as *implication relation*, between formulae is used at multiple places in the architecture. We give some important applications of reasoning in the architecture next.

9.3.12.1 Application to Matching Requests against Data

When a party (e.g., user) wants to fulfill a data request ψ of another party, the party needs to find a combination of formulae of its identifier or identity relationships such that the sequent $\phi_1, \ldots, \phi_k \vdash \psi$ holds, with the formulae $\phi_1, \ldots, \phi_k$ being from the party's identity or identifier relationships. Note that an environment $\mathcal{E}$ defines the instantiation of all free variables in ψ. The environment is closely related to the choice of formulae $\phi_1, \ldots, \phi_k$. We give details on the matching of a request with identity relationships further below in this chapter.

9.3.12.2 Application to Evaluation of Authorization Policies

Another interesting use case for derivations over our logic is the evaluation of authorization policies. As a specific step during the evaluation of an authorization policy rule, the policy engine needs to check whether the subject and object expressions of the rule are fulfilled, given the information available about the subject (requester) and object (policy target). Thereby, the left side of the sequent are the formulae comprising information received about the requester, the right side is the formula comprising the (instantiated) subject and object expressions. The tight integration of the policy model with the logic-based data model allows for powerful expressiveness in the definition of policies as well as use of derivations over the data model within the policy evaluation algorithm. See Section 9.7 for details.

9.3.12.3   Application to the Abstract Expression of Certifiers

As another interesting example for reasoning, consider the abstraction of the specification of certifiers which facilitates the openness of the system in terms of flexibility in addressing certifiers within policies by more abstract means than merely referring to them as identified parties.

$$
\begin{aligned}
o_1 = &\forall \mathsf{Cid}, \mathsf{U} : Eq(\mathsf{Cid}.country, \mathsf{Switzerland}) \wedge \\
&Eq(\mathsf{Cid}.certifiertype, \mathsf{Governmental}) \wedge Eq(\mathsf{Cid}.subject, \mathsf{U}) \wedge \\
&Reputation(\mathsf{U}, 10) \rightarrow \\
&Trustlevel(\mathsf{U}, 10)
\end{aligned}
$$

The example shows how the truth of a predicate $Trustlevel(\_,\_)$ can be derived from information about a certifier's identity as well as other predicates.

Consider furthermore that the following fact P expressed as a predicate holds and that the given formula ψ is true.

$$
\begin{aligned}
P = &Reputation(\mathsf{u}, 10) \\
\psi = &Eq(\mathsf{cid}.country, \mathsf{Switzerland}) \wedge Eq(\mathsf{cid}.certifiertype, \mathsf{Governmental}) \wedge \\
&Eq(\mathsf{cid}.subject, \mathsf{u}) \wedge \ldots
\end{aligned}
$$

Now the following sequent can be proven to hold based on the assumptions:

$$
P, \psi \vdash_{o_1} Trustlevel(\mathsf{u}, 10) \tag{9.1}
$$

Reasoning similar as in the example is very useful in stating policies more abstractly than by identifying the certifier or set of certifiers that are accepted for a given identity. A subject expression in a policy building on such ideas could contain the following:

$$
Eq(\mathsf{C}.lastname, \mathsf{Lastname}) \wedge Eq(\mathsf{C}.certifier, \mathsf{U}) \wedge Trustlevel(\mathsf{U}, 10))
$$

A major challenge for a large-scale use of these ideas of reasoning is that people should be able to obtain and agree on the ontologies to use. The first challenge requires that there be providers of ontologies that are trusted by both partners in an interaction. Possible such parties can be independent data protection authorities such as the German ULD[7] whose primary concern is the data protection of users. The second challenge is that two parties need to agree on an ontology to use for their reasoning within an interaction in order to leverage the power of automated derivations in the logic. The issue of interaction partners agreeing on ontologies as well as an overall ontology architecture for identity management has been addressed in [HS06].

---

[7] http://www.datenschutzzentrum.de

9.3.13   Requests of Data

For our architecture, we need to express requests of data and other resources in addition to data statements. Such requests are, for example, communicated during a negotiation protocol between two parties in order to request data or other resources from the respective other party.

Requests are modeled in a similar way to data statements discussed earlier in this section, though with some important differences. A main difference is that instead of referring to identities with individual constants, free variables are used. Attribute values are requested by specifying a predicate expressing equality between the attribute of the identity and a free variable instead of a constant representing the attribute value as in a data statement. Furthermore, parties are referred to through variables instead of through constants. It is an integral part of the agreed processing that all free variables need to be instantiated with concrete terms in a satisfying response to a data request.

Example (Data request): The following example requests the *firstname* attribute as well as a proof that a predicate holds over the monthly salary of the party, both based on the same identity of type Bank_Statement. The request requires that both be proved using a specific variant of the Identity Mixer private certificate system. Note the free variables C standing for the identity as well as Firstname for the value of the requested attribute.

$$Eq(\mathsf{C}.firstname, \mathsf{Firstname}) \wedge Geq(\mathsf{C}.monthlysalary, 3500) \wedge$$
$$Eq(\mathsf{C}.currency, (EUR)) \wedge Eq(\mathsf{C}.type, \mathsf{Bank\_Statement}) \wedge$$
$$Eq(\mathsf{C}.protocolsuite, \mathsf{Identity\_Mixer\_2048\_bit}) \wedge$$
$$Eq(\mathsf{C}.certifier, \mathsf{u}) \wedge \ldots \qquad \qquad \square$$

A reponse to such a request needs to follow the rule that each of the request's free variables needs to be instantiated in the response formula. The free variables are C representing an identity as well as Firstname representing the attribute value of the *firstname* attribute of C. The following is a proper response:

Example (Minimal data response):

$$Eq(\mathsf{bs}.firstname, \mathsf{Jane}) \wedge Geq(\mathsf{bs}.monthlysalary, 3500) \wedge$$
$$Eq(\mathsf{bs}.currency, \mathsf{EUR}) \wedge Eq(\mathsf{bs}.type, \mathsf{Bank\_Statement}) \wedge$$
$$Eq(\mathsf{bs}.certifier, \mathsf{u}) \wedge \ldots \qquad \qquad \square$$

So is the following, exposing more information about the salary, but still fulfilling the formula:

Example (Data response):

$$Eq(\mathsf{bs}.\mathit{firstname}, \mathsf{Jane}) \wedge Eq(\mathsf{bs}.\mathit{lastname}, \mathsf{Doe}) \wedge$$
$$Geq(\mathsf{bs}.\mathit{monthlysalary}, 6000) \wedge Eq(\mathsf{bs}.\mathit{currency}, \mathsf{EUR}) \wedge$$
$$Eq(\mathsf{bs}.\mathit{type}, \mathsf{Bank\_Statement}) \wedge Eq(\mathsf{bs}.\mathit{certifier}, \mathsf{u}) \wedge \ldots \qquad \square$$

A request for proving holdership of a pseudonym can be formalized as follows in our language, making use of the identifier variable P and its attributes:

Example (Request for proving identifier holdership):

$$Eq(\mathsf{P}.\mathit{subject}, \mathsf{Subject}) \wedge Eq(\mathsf{P}.\mathit{subjectId}, \mathsf{SubjectId}) \wedge \qquad \square$$

Note that variable names should be chosen according to a scheme to avoid accidental reuse of them and that there are rules how to choose terms referring to identities in the response, depending on the used protocol.

Through having variables instead of constants in multiple places of a request formula, it represents the *class* (set) of all formulae that can result by instantiating the free variables with constants. Thus, a data request is equivalent to a class of formulae with the operational semantics that a concrete instance needs to be chosen by assigning constants to the free variables. Each formula that allows for deducing the request with its variables instantiated properly is a valid response to the request. For example, a request asking for a proof based on a salary statement that the salary of a user is greater than or equal to 1500 EUR can be answered in various ways: the most data-minimizing response is simply a proof that the predicate asked for in the request holds; less data minimizing statements are ones that reveal that the salary is greater than or equal to some value which is itself greater than or equal to 1500, consistent with the salary statement the user holds; the most-revealing statement is the one that simply reveals the attribute value that must be greater than or equal to 1500.

Requests targeted at the interaction partner and asking for data about it or other parties are processed by the interaction partner against its identifier and identity relationships (and possibly other data) by using the usual means of matching the party's formulae with the request as outlined in this chapter.

Requests of data are expressed in the subject and object expressions of policy rules as explained in detail in Section 9.7. When a resource is requested, multiple such expressions of all applicable policy rules may be composed to a single data request sent to the party. Furthermore, such requests can be created at the application layer, e.g., by a service provider, to request data about a party from another service provider.

9.3.13.1   Third-Party Requests

The default data provider to answer (parts of) a request is the interaction partner. Though, there are valid cases where the data about the interaction

partner are to be requested from a third party during an interaction. Take as an example a user whose authorization policies require that the interaction partner have a certain minimum reputation score as stated by a reputation provider the user trusts for the purpose of providing this information. In this case, the request needs to specify information about the data subject of the to-be-requested data such that the third party can return the requested data. Such a request is called *third-party request*. Specifying the data subject is done by specifying relevant information in the request, e.g., a commonly-known or public identifier of the subject of the request, as in the following example. For subjects with a publicly-known identifier such as most service providers in today's Internet, the use of such an identifier is the simplest approach as there is no need for an additional agreement of a new identifier between the party and the third party for the subject.

Example (Request formula with identity specification):

$$Geq(\mathsf{R}.reputationscore, 8) \land$$
$$Eq(\mathsf{R}.uniquename, \mathsf{My\_Electronics\_Shop\_Co})$$

□

The predicate over the *reputationscore* attribute needs to be answered in the response, that is, the given predicate in the example expresses the usual request semantics for attribute information, while the attribute *uniquename* is provided in its value and used for identifying the party about which the reputation is requested. Using the concept of *sanitizing policies* that can protect sensitive information contained in policy rules on the side of the requesting party allows for transforming this request into a request for the value of the reputation score before being sent to the third party, thereby hiding the predicate expressed over the attribute value. This prevents the subject expression, the *Geq*-predicate and constant 8, of this policy rule of the user from leaking to the third party. In this case, the following request would be sent to the third party:

$$Eq(\mathsf{R}.reputationscore, \mathsf{Reputationscore}) \land$$
$$Eq(\mathsf{R}.uniquename, \mathsf{My\_Electronics\_Shop\_Co})$$

It is also possible that a request comprises an instantiated identity term instead of a variable for an identity. This is the case when the requesting party and third party share a term for the identity, e.g., when a service provider requests data about a user from another service provider while they share a subject identifier of the related identity. Due to the established linkability, this approach is not suitable for interactions where interactions should remain unlinkable.

$$Eq(\mathsf{R}.subject, \mathsf{service68}) \land Eq(\mathsf{R}.reputationscore, \mathsf{Reputationscore})$$

The example can be used to query the reputation score of a party by addressing the party through its subject identifier the reputation provider uses in communication with the party. The identifier may take on any of the terms that are used between the parties to communicate about the subject.

9.3.13.2   Processing of a Third-Party Request

When a third-party request arrives at the third party (e.g., the reputation provider), it is processed as a resource request and an instance of a negotiation protocol is executed. Once successfully completed, the resource access is processed. The parts of the request that specify the subject of the data are used to retrieve the profile record related to the subject, containing the requested data. The request in its whole defines the parts of the record being accessed (read). For the example above, this requires that the requesting party needs read rights for the attributes specifying the name and reputation score of the subject the request is about at the third party in order for the third party to return the response to the request. The authorization of the request is done through the mentioned negotiation protocol.

This approach of expressing the request for data about parties has similarities to a query as implemented, for example, in SQL, in its meaning, though, our language is a specifically-designed language targeted at handling identity information. Unlike SQL queries, our language is capable of expressing disjunctions in a simple way which is an important aspect for data minimization realized through disjunctions.

Checking whether one or more formulae match such a request is done as usual by checking whether the implication relation holds, with the difference that the free variables must be instantiated beforehand consistent with the attribute statements of the formulae against which the matching is done. See Section 9.3.14 on the formalism behind this processing. Note that exactly the same reasoning in our logic is used as in the case of matching a request with formulae without any identity information being specified. Suitable answers to a request are based on such identities of the reputation provider that match the request. In the reputation example, changes in the reputation are reflected in new identities being generated by the reputation provider and the latest one being used to answer the request.

9.3.14   Matching Data against Requests

Matching of data against a request refers to the process of a party matching data formulae it holds, particularly such of identity and identifier relationships, against a request from another party. Such matching needs, for example, to be performed within the negotiation protocol when processing a request of the other party in the interaction; see Section 9.9 for details on the negotiation protocol. In such a protocol, the request is built from the subject and object

expressions of authorization policy rules that apply to a resource of the other party that is accessed by the party during the protocol.

Let ψ be a request by the other party in an interaction, with the free variables to be instantiated with constants. Let $\phi_1, \ldots, \phi_n$ be all the formulae of identity and identifier relationships (or other formulae) to be used in the matching process. The output of the matching is a set of results. Each result is a set $\Phi_i = \{\phi_{l_1}, \ldots, \phi_{l_{k_i}}\}$ of formulae and an environment $\mathcal{E}_i$ such that $\phi_{l_1} \in \Phi_i, \ldots, \phi_{l_{k_i}} \in \Phi_i \vdash^{\mathcal{E}_i} \psi$. Furthermore, the list of formulae on the premises side of the sequent is such that it is minimal, that is, no formula can be removed from the list without making the sequent not fulfilled. That is, no formula is there without it being needed in order to fulfill the request formula. Note that an ontology $\mathcal{E}$ can additionally be used as outlined in Section 9.3.12 to express knowledge.

We do not present a possible matching algorithm in detail, but rather sketch the basic ideas of a simple, yet practical, algorithm for this purpose. The algorithm is not optimized, though it is sufficiently fast for practical applications of privacy-enhanced IdM. The algorithm needs to determine the matching formulae from the list of input formulae for each identity and identifier variable of the request. A matching formula for an identity is one in which an identity is referred to that is a valid instantiation of the variable standing for the identity in the request, regarding the identity type, its attributes, certifier, protocol suite, and other metadata attributes. This holds analogously for identifiers. For each matching formula, the variables of the request representing the identity or identifier and its uninstantiated attributes need to be instantiated based on the matching identity or identifier, that is, using the attribute values of the matching object. The instantiation of the free variables becomes a part of the environment once the result is composed as explained below. Disjunctions in the request require additional consideration as it is sufficient to satisfy one of the sub-formulae of a disjunction with formulae the party holds. Once all matching formulae have been found for all identity and identifier variables in the request, all their valid combinations can be constructed together with the environments such that the sequent shown further above holds for each such combination of formulae with the corresponding environment. For each valid combination of formulae and the corresponding environment, that is, each result $(\Phi_i, \mathcal{E}_i)$, a single formula ψ' can be created that is the instantiation of the request ψ based on the environment and the matching formulae. Any of those formulae could be released by the party to fulfill the request using the appropriate data release protocol based on the identity relationships that the matching formulae pertain to. Of all those formulae, the best-suitable one, e.g., in terms of data minimization and avoiding linkability with previous interactions and minimizing collateral release, is chosen. The chosen formula can then be released to the interaction partner through the data exchange component explained in Section 9.6.

Special consideration needs to be given to conditionally-released identities and opaque identities. We define the matching semantics such that such identities always match with on-the-fly generated identities with attribute values as specified through the request formula and the objects those identities are related to.

The formulae $\phi_1, \ldots, \phi_n$ are all formulae of the party's identity relationships, formulae of identifier relationships of which the communication partner is the party with which those have been established, or profiles[8]. That is, the data considered in the matching comprises all data for which the party can run data release protocols for data endorsed by third parties or identifiers with the other party. One can think of scenarios also when a party would need to reveal data about other subjects than herself in an interaction; in such cases, for example, also formulae from the profile data may be part of the formulae to match against. The same is important for service providers who release data about their customers to other service providers or reputation providers who release data about parties they make reputation statements about. The list of formulae used in the matching thus depends on the setting being considered. The matching process is the same regardless of whether a formula belongs to an identity or identifier relationship or a profile data record – the attributes and the certification metadata ensure that only suitable formulae match with a request. In case the party is a user, the human may be involved in the process of deciding which of the result formulae of a matching process should be released in the end, e.g., depending on the pseudonym under which the data is to be released or which identity relationships should be used, and in consenting to the release.

For a real-world deployment, fragments of both the data model and the matching functionality can be implemented, thus leading to a less expressive but simpler system. For first practical deployments of user-centric identity management systems based on private certificate systems as the data release protocols, a first deployment phase that does, for example, not support disjunctions in a data statement, is reasonable as a first step for a deployment of private certificate systems.

9.3.15   Further Discussion

We next discuss some additional aspects related to our data model to provide more details of selected aspects to the reader.

Operators on Data

We define the binary operator $d_1 \preceq d_2$ on two formulae as the operator that compares two formulae and that returns true on its arguments if and only if d_1

---

[8] At the time of executing this matching process, at least some identifier of the other party should be known in order to find potentially matching identifier relationships.

is a sub-formula of d_2. Similarly, we define the operator $d_1 \prec d_2$ to represent the proper subformula relation between the two formulae. Analogously we define the relations for a formula being a super-formula of another formula using the $d_1 \succeq d_2$ operator and $d_1 \succ d_2$ for a formula being a proper superformula of another formula.

Kinds of Identities in a Formula

An identity formula usually has multiple identities being referred to through its predicates, concretely this being the *core identities* that are used for making the actual identity statement, identities with certifiers of the further identities being the subjects, identities with conditional data recipients for conditionally-released identities being the subjects, and opaque identities. In a rather complex formula, this requires a formal, that is, machine-computable, approach of identifying the identities that are the basis for the data statements and the ones that comprise the metadata of the formula. Algorithmically, this can be achieved as follows: Every identity of a formula that is of type identity and that is not an identity with the subject being one of the certifying parties expressed in the formula are such core identities of the formula.

Expressing Holdership of an Identity

Showing the fact of holding an identity of a specific identity type without making any statements over its data attributes is a frequently-required operation in our view as many permissions can be expressed through holdership of an identity of a certain type certified by a specific party. This can be expressed by specifying the type of the identity and possibly further metadata contained therein such as the certifier and the temporal validity of certification. This integrates well with our model without the need of introducing a new language concept, because we have the type and other metadata expressed as attributes of identities. Expressing holdership of identity usually requires that at least the type of the identity needs to be specified as otherwise any identity would be suitable and thus insufficient information about the identity would be available for an authorization decision.

Example (Possession of an electronic Swiss passport): The following example is a statement showing the possession of an electronic Swiss passport without revealing any of its data attributes. The system-defined macro today gets expanded to the current date within the protocol for releasing the formula to the other party.

$$Eq(\mathsf{p}.type, \mathsf{Swiss\_Passport} \wedge Geq(\mathsf{p}.validuntil, \mathsf{today}) \wedge$$
$$Eq(\mathsf{p}.certifier, \mathsf{u}) \wedge \cdots \qquad \qquad \square$$

Take as another example a party that proves possession of an entrance ticket for entering a dance club: Only the type, certifier, and temporal validity of

the entrance ticket can be sufficient for a decision on the granting or denial of entrance to the club. When thinking of a widespread use of an electronic identity system as ours, many more examples of similar cases of authorization policies come to one's mind.

Expressing the Communication Partner

Another predicate useful for interactions between parties is the predicate $CP(\cdot)$ expressing that a party is the communicating party in an interaction, communicating with the other party. This predicate allows one to differentiate between parties (subjects) in an identity statement, e.g., for a user-centric delegation use case where a distinction might need to be drawn between the delegater and the delegate. We only refer to this predicate in our discussions when it is required, otherwise, we leave it out for simplification of the discussion.

$$CP(subject, \textsf{session})$$

Terms and their Naming

An identity can, in our model, be referred to by using distinct terms or variables at different places when talking about it. This is an extremely useful property of our data model as it allows a party to refer to one of its identities they hold, e.g., one backed by a private certificate, using a different name every time they use it. This is an important property for preserving anonymity when using a single identity multiple times in interactions with other parties, thus this feature is the chosen means for avoiding the introduction of undesired linkability between data release interactions.

Further Discussion about Identities.

An identity technically pertains to exactly one *subject*, the subject of the data represented through the identity and the party indicated with the *subject* attribute of the identity.[9] An identity is also associated with at most one certifier, that is, an entity (e.g., a single party or a logical party comprising multiple physical entities) who vouches for the identity. An identity may contain an attribute for representing the subject under which it is known by the certifier. This identifier is the identifier of the subject with the certifier and can be used for allowing for the revocation of the anonymity of a data release interaction.

---

[9] The concept of subject correspondsto the concept of *data subject* of the European data protection legislation in case the subject is a user and only data about a single person is represented in the identity; this legal term does not apply in case the subject is a service provider. Note that we use the term in its technical, that is, more general, meaning in this chapter unless explicitly stated otherwise. Even if an identity would comprise data about multiple persons, the indicated subject is responsible for proper handling of the identity.

A formula over multiple identities may make statements over identities with different subjects, that is, talk about different parties which can, for example, be useful in scenarios with user-centric delegation where one needs to express data about the delegater and the delegatee within a single formula.

Another view of identities being part of identity relationships is that such an identity is a view the certifier of the identity has of the subject and that it has decided to vouch towards other parties. This represents well what an identity actually is, namely what a party who makes a statement about another party states about this party. Thus, it is not at all about actual correctness of the attributes with respect to any real-world "reference" attributes, e.g., the ones in a party's credentials such as their passport, rather it is about claims someone, who is trusted by others for this, makes about someone else. Depending on the claiming party and its policy of validating attributes it vouches for, those claims may be suitable for practical purposes in identity management for other parties, such as service providers relying on those claims. Actual correctness of the attributes with respect to the real attribute values, such as the civil identity of a user, is then not a technical question, but a question of processes executed and verification performed by the certifier as well as communicating this.

Implementation

In our prototypes, experiments have been performed of modeling and implementing a previous, and less powerful, variant of our data model based on W3C's RDF and OWL languages because of the existing tools for those. In our current presentation of the architecture we do not fix the ideas to any specific technology and thus use first-order logic in this chapter.

9.4 Data Representation Based on Our Model

In this section we discuss the application of our data model to the different classes of data that a party handles as introduced in Section 9.2.3, more concretely the identifier relationships, identity relationships, the data track, as well as the profile data. We discuss how the data model is used to express those data held by a party and give examples to illustrate the application of the data model.

Additionally to the mentioned kinds of data, one needs to represent also the following using a formal data representation: Requirements of requesters as expressed in access control policies as well as the policy target and requests and data statements communicated between parties for which we also employ our data model.

As we will see in this section, each of the different kinds of data is modeled in essentially the same way, including the association of metadata with the data: Each class of data is represented through a set of tuples, each comprising

a record and a set of metadata predicates on the record. Each record is again a set of tuples, each tuple comprising a formula in our fragment of first-order logic as explained in Section 9.3 and a set of metadata predicates expressed on the formula.

For the specification of metadata, we only present the most prominent metadata items; further predicates than the ones presented can be defined in practice if this is required in a particular setting.

9.4.1   Identifier Relationships

The identifier relationships Π of a party represent identifiers a party has established, each one being about a party, be it itself or another party, for use in communication with another party or multiple other parties. The set Π is a set of identifier relationships ν_i comprising both the data and metadata of the identifier relationship where the data are represented using our data model as outlined in this subsection.

An identifier relationship ν is a tuple comprising an element ν' and a set of metadata predicates associated with the element. The element is a set of cardinality 1 of a tuple ν'' comprising a formula ϕ specifying an *identifier object* p through its subject and the subject identifier value and a set of metadata predicates $\{m_j\}$ on the formula.[10] Below we give the basic structure of a formula ϕ representing an identifier relationship. The term s is used to refer to the subject of it, the constant sid is the actual identifier. In an interpretation of the formula, the *subject* will be interpreted with the party, the *subjectId* with itself.

$$\phi = Eq(p.subject, s) \wedge Eq(p.subjectId, sid)$$

An identifier relationship ν contains a metadata predicate specifying the subject term, that is, the party to whom the identifier relationship applies. The following metadata are associated with a formula: the subject identifier of the party with whom the relationship has been established; the mapping from the locally-used terms to the terms used in communication with the other party; cryptographic material (e.g., cryptographic pseudonyms) for using the identifier relationship; and further metadata. An identifier relationship can be obtained by the party by executing an appropriate protocol with another party or by declaring it locally through the console.

When a matching of locally-held data with a data request issued by a communication partner is to be performed, the party retrieves all identifier relationships that have the subject identifier of the other party as metadata. The formulae of all those identifier relationships are used as input to the

---

[10] The identifier relationship is modeled with a set of tuples of formulae and metadata with a restriction that the set be of cardinality 1 in order to use the same modeling as for the other kinds of data. The restriction on the set applies as for identifier relationships only one formula is required to be modeled.

processing. If multiple ones fulfill a part of the data request, the choice is computed by the party's preferences or by soliciting input from the human user in case the party is a user.

9.4.2   Identity Relationships

Identity relationships are one of the most interesting data entities of a party to consider as they are one of the foundations of user-centric privacy-enhancing data exchange which is central to this work. Thus, they receive particular attention in our discussions. A party holds a set Γ of identity relationships. An identity relationship γ is a tuple of an element γ' and a set of metadata predicates on γ'. The element γ' is a set of tuples γ'' each comprising a formula ϕ and a set of metadata predicates on the formula. That is, an identity relationship can have one or more data formulae $\phi_1, \ldots, \phi_l$ making statements over identities as discussed in the section on the data model. One formula—the *core formula*—specifies exactly the data the certifier of the identity relationship vouches for. The other formulae may comprise less information on the identities and it must hold that the core formula implies every other formula in the logic (i.e., $\phi \vdash \phi_i$ for all ϕ_i). Additional metadata associated with the identity relationship and the formulae can express information required for using the identity relationship.

9.4.2.1   Data Formula

A formula ϕ expresses both the identity data and parts of the metadata on the identity data in an integrated way by using our data model. The formula makes statements about at least one identity, the *core identity* of the identity relationship.[11] This is the identity comprising the attributes the identity relationship is about, additional identities, typically one, may be used to specify the certifier. Thereby, the core formula expresses exactly the data that the certifier of the identity relationship vouches for. The permitted syntax is governed by the protocol underlying the identity relationship. The formula must not contain free variables as it has the meaning of being a concrete formula expressing data and not a class of formulae as in a request.

   We next give an example of a formula ϕ that determines the data being vouched for in an identity relationship:

---

[11] Practically, it is sufficient to have a single identity of this kind in an identity relationship, though the architecture and processing can support multiple if this should be required.

$$\phi = Eq(\mathsf{dl}.\mathit{subject}, \mathsf{user4}) \land Eq(\mathsf{dl}.\mathit{type}, \mathsf{Swiss\_Driver's\_License}) \land$$
$$Eq(\mathsf{dl}.\mathit{firstname}, \mathsf{Jane}) \land Eq(\mathsf{dl}.\mathit{lastname}, \mathsf{Doe}) \land$$
$$Eq(\mathsf{dl}.\mathit{dateofbirth}, \mathsf{1977\text{-}12\text{-}12}) \land Eq(\mathsf{dl}.\mathit{vehicleweight}, \mathsf{3000}) \land$$
$$Eq(\mathsf{dl}.\mathit{protocolsuite}, \mathsf{Identity\_Mixer\_2048bit}) \land$$
$$Eq(\mathsf{dl}.\mathit{certifier}, \mathsf{germanGovernment}) \land$$
$$Eq(\mathsf{cid}.\mathit{subject}, \mathsf{germanGovernment})$$

The formula ϕ expresses that the subject of the identity dl is user4, its type is Swiss_Driver's_License, the *firstname* attribute in the identity dl has value Jane, the *lastname* attribute has value Doe, and the *dateofbirth* attribute has value 1977-12-12, that the protocol suite being used for the identity relationship is Identity Mixer with 2048 bit, and that the certifier of identity dl is party germanGovernment with germanGovernment being a constant term. The identity further specifying the certifier is cid. The formula ϕ above does not further specify any properties or identifying attributes of the certifier as it is assumed that cid is a public identity that can be retrieved by the party. The *subject* attribute refers, by our convention, to the party using a term that is locally used by the party to represent itself. By convention, always the same name is used locally.

For a specific identity relationship, the underlying protocol determines the expressiveness that may be used for specifying the formula. The Identity Mixer private certificate system allows for a single core identity to be referred to, conjunction as only logical connective, and the predicates Eq, Neq, Geq, Gt, Leq, and Lt for relating attributes to their values.

9.4.2.2   Certifier Specification

In addition to ϕ, the party holds—or obtains in a sub-protocol—another formula ψ in a profile data entry for germanGovernment that makes statements about the identity cid of the certifier germanGovernment. The following example is a continuation of the one above and represents the information about the certifier as follows through ψ:

$$\psi = Eq(\mathsf{cid}.\mathit{subject}, \mathsf{germanGovernment}) \land$$
$$Eq(\mathsf{cid}.\mathit{uniquename}, \mathsf{Swiss\_Motorvehicle\_Administration}) \land$$
$$Eq(\mathsf{cid}.\mathit{protocolsuite}, \mathsf{X509})$$

This formula makes an identity statement about the certifier using the same term germanGovernment for referring to this party as in ϕ. Over the identity cid identity statements can be made about the party germanGovernment as usual for formulae specifying identities. The formula ψ is stored by the party as part of the profile data, namely as part of the profile with subject being the certifier germanGovernment.

In the typical case, the certifier identity cid will contain identifying attributes of the party referred to by its subject because anonymity is not required for such a party. Through using the same term germanGovernment for referring to the certifying party in both formulae ϕ and ψ above, the identity cid specifying the certifier is related to the identity dl being described in the identity relationship within the data model as being one of its certifier's identities. It is an inherent property of our data model that the same subject term refers to the same party throughout formulae; this is a crucial property in the above example. In most cases, the certifier identity cid is public and referred to by the same term by all parties. For the currently not so prominent case of the certifier remaining anonymous or pseudonymous without being identified, it will appear under fresh identifiers towards different parties it issues identity relationships to.

When multiple identities of identity relationships have the same party as certifier referred to through the same subject term—germanGovernment in the example above—the formula for the certifier needs to be specified only once in the profile data of the party as it always describes the same identity of the same certifier. Section 9.3 explains how the matching of identity relationships with a data request is done for computing formulae satisfying the data request using the data model. This already takes care of considering the certifiers if properly referred to in the formulae and represented at the party.

9.4.2.3   Multiple Data Formulae and Restrictions

Multiple formulae $\phi_1, \ldots, \phi_k$ can be associated with the same identity relationship γ. One formula, denoted as the core formula and tagged through metadata as such, must specify exactly what the certifier vouches for. Additional formulae may be used to specify a fragment of the data of the identity relationship in order to associate a different authorization policy with this fragment. The core formula must imply each of these other formulae. A common example for this is that the core formula, w.l.o.g. referred to as ϕ_1, specifies a value for a specific attribute of the identity while an additional formula ϕ_2 specifies one or more predicates over the attribute. Then, a restrictive authorization policy can be associated with the attribute value in ϕ_1 while a less restrictive authorization policy can be defined on the predicates over the attribute in ϕ_2.

Example (Multiple data formulae):   With this example we show the use of an additional formula of an identity relationship and introduce the concept of a *restriction*. Let the formula ϕ_2 define a predicate on the attribute *monthlySalary* of the bank statement identity b expressing that the attribute is greater than or equal to the variable Lowerbound. Let furthermore the restriction ρ_{ϕ_2} on ϕ_2 specify predicates on the range of this variable.

$$\phi_2 := Geq(\text{b}.monthlySalary, \text{Lowerbound})$$
$$\rho_{\phi_2} := Leq(2500, \text{Lowerbound}) \wedge Leq(\text{Lowerbound}, 4000)$$

The above example formula ϕ_2 and its restriction ρ_{ϕ_2} define that the *salary* attribute of the identity is greater than or equal to the variable Lowerbound where the latter ranges—specified through the restriction—from 2500 to 4000, both inclusive. A different, e.g., less restrictive authorization policy than the one on ϕ can be defined on the formula ϕ_2 as it reveals much less information.   □

The matching of ϕ_2 with its restriction against a data request in our logic needs to consider the formula ϕ_2 as well as the restriction formula ρ_{ϕ_2}. Both need to be true under the chosen assignment of the variable Lowerbound and considered in the choice of values in the matching algorithm. The restriction never becomes part of the response to a data request, but is only used for finding suitable responses.

As a special case of the previous example, the following formula defines the release of partial information on the attribute with a constant for the predicate and without the flexibility of using a restriction.

$$\phi_3 := Geq(\mathsf{b}.salary, 3000)$$

The approach of specifying ranges to be revealed fits, as described, well into the overall formalism without major extensions of the data model. Authorization policies and data handling policies can be associated with each formula $\phi_\_$ of the identity relationship which is the reason why we allow for formulae in an identity relationship in addition to the core formula.

In addition to ranges over the integers or other attributes over totally-ordered sets, a good example is that the country of residence attribute in an electronic passport may be released to be in the set of all EU member states without restrictions on the use of it; if the value of the country of residence is to be revealed, though, a stricter access control policy or data handling policy may be required to be enforced, e.g., the potential recipient would need to prove holdership of a certification by a data protection organization of applying a minimum standard regarding its privacy practices. Set membership $a \in S = \{a_1, \ldots, a_l\}$ can be expressed by a disjunction $Eq(a, a_1) \vee \ldots \vee Eq(a, a_l)$ for all $a \in S$ in FOL.

We observe that *information cards* in the CardSpace model of identity management are conceptually similar to the identity relationships defined in our work. They are expressed in XML while our model is defined over a formal logic with all its advantages as discussed in this work. A main strength of our identity relationships is their embedding into the overall formalism and expressivity provided by our data model and thus integration into our architecture. Our approach also allows for general ways of associating policies with the data of the identity relationship or selected parts of it.

9.4.2.4   Use

The identity relationship γ can be used by its holder to release any formula θ that can be derived from any of its formulae $\phi_\_$ to a data recipient. The more

general case is that multiple formulae $\phi_1, \ldots, \phi_k$, also from different identity relationships and identifier relationships, can be used as a basis for deriving a formula θ. The formula θ can thereby comprise parts of the information of the formulae it is derived from. The necessary precondition is always that the (cryptographic) protocols underlying the identity relationship can be used to prove θ correct. See Section 9.3 for the formalism behind deriving a formula from a set of formulae.

The following data formula θ is a possible result of using the above identity relationship to release data based on it:

$$
\begin{aligned}
\theta = &\ Eq(\mathsf{dl45}.subject, \mathsf{user53}) \wedge Eq(\mathsf{dl45}.type, \mathsf{Swiss\_Driver's\_License}) \wedge \\
&\ Geq(\mathsf{dl45}, vehicleweight, 1000) \wedge \\
&\ Eq(\mathsf{dl45}.protocolsuite, \mathsf{Identity\_Mixer\_2048\_bit}) \wedge \\
&\ Eq(\mathsf{dl45}.certifier, \mathsf{germanGovernment}) \wedge \\
&\ Eq(\mathsf{cid}.subject, \mathsf{germanGovernment})
\end{aligned}
$$

This formula θ comprises parts of the information of the formula ϕ above it has been derived from: it reveals parts of the attributes of ϕ by their values and states that the attribute *vehicleweight* be greater than or equal to 1000, but does not reveal its value. Such capabilities of partial release of information are supported by a private certificate system underlying the identity relationship. The above formula is a typical example for data minimization in identity management, which is a core topic of our work.

When following the convention that an identified certifier is always referred to by the same term for the subject as well as the related identity is always referred to by the same term, there is no need to additionally include the sub-formula specifying germanGovernment through its identity cid. A data recipient can retrieve this information, e.g., through public key certificates. As an alternative, it can also be sent within the protocol for establishing the identity relationship.

9.4.2.5   Metadata

An identity relationship and its formulae need metadata associated with them in order to be utilized for releasing data in interactions with other parties. We give an overview of the most important metadata of identity relationships and its formulae next.

The metadata comprises cryptographic material, such as private certificates or public keys if they are stored directly within the identity relationship. We also need a metadata predicate that specifies whether an identity relationship is one the party is holder of or whether it is one the party vouches for. If we do not specify this explicitly in this work, we assume that it is clear from the context. Another predicate expresses whether the identity relationship is still active or has been deactivated, e.g., because it has expired, been revoked,

or been superseded due to a change of attribute data. The core formula of the identity relationship is flagged through a metadata predicate as such. The metadata must express everything that is (technically) needed for the identity relationship to be utlized in interactions for releasing data and that is not contained yet in the data formula. For an identity relationship based on a private certificate, this particularly comprises the certificate structure, a technical specification of the internals of private certificates of a specific type. See Section 9.6.6.1 for details.

9.4.3   Data Track

The data track Δ of a party models data statements that have been released to other parties as well as associated metadata. The idea is that each release of identity-related information that the party's system is aware of is recorded in the data track.

A party's data track Δ is a set of tuples δ, denoted data track entries or data track records, where each tuple comprises an element δ' and a set of metadata predicates on this element. Each element comprises a set of tuples δ'' of the form $(\delta_k^*, \{m_{k,l}\})$ of a formula δ_k^* over our data model and a set of metadata predicates associated with the formula.

A single data track record δ comprises data related to a single recipient. Multiple data releases by the party within a single session and even within multiple sessions with the same recipient can be captured by one record. Formulae in one data track record can refer to different subjects, e.g., the party and certifiers, though, and each record is associated with the main subject the data is about.

A formula represents a data statement that has been released to or obtained by another party, expressed in our data model. Metadata can be expressed as for other kinds of data on the data track record as well as on each formula of a record in order to store relevant information.

9.4.3.1   Data Formula

Each formula of a data track record is a formula that has been previously released to or otherwise obtained by another party that has been captured in the data track after the data release interaction. The formula contains the identity data as well as metadata sent to the other party.

As an example of a formula based on the identity relationship shown before, that is, with renamed subject and identity terms of the identity relationship used, take the following:

$$\phi'' = Eq(\mathsf{dl45}.subject, \mathsf{user53}) \wedge Eq(\mathsf{dl45}.type, \mathsf{Swiss\_Driver's\_License}) \wedge$$
$$Geq(\mathsf{dl45}.vehicleweight), 1000) \wedge$$
$$Eq(\mathsf{dl45}.protocolsuite, \mathsf{Identity\_Mixer\_2048bit}) \wedge$$
$$Eq(\mathsf{dl45}.certifier, \mathsf{germanGovernment} \wedge$$
$$Eq(\mathsf{cid}.subject, \mathsf{germanGovernment})$$

The party stores the formula with the renamed terms, that is, exactly the formula as released to the other party. This immediately shows linkabilities that are explicitly established through identifiers between different interactions with the other party and does not introduce non-existing linkabilities in the view of the party as would be the case when using the local terms of the party for referring to parties and objects. As the mapping of terms is available to the party, it can always obtain the terms it uses itself for addressing parties and objects.

9.4.3.2   Metadata

Metadata on a data track element δ' comprises an identifier of the recipient of the data, that is, the party whom the record is associated with. The convention for the choice of an identifier of the recipient is the *subject* term of the party the formula has been released to. For each such subject there exists a profile data record with the same identifier which may contain information about the other party data has been released to, unless in the case that data has been released to a party that has not provided any data about itself.[12]

The subject of the released data may be represented already in the formula specifying the data and is also stored as a metadata predicate. Note that the subject is often times the party itself, but can equally be another party in case the party releases data about other parties as may be the case for a service provider who releases data of their customers to a business partner or a user who acts for another party under a delegation relationship.

Metadata predicates expressed on each data formula of the data track record comprise the following: the policy under which the data has been released (through its name or copy of the policy), identity and identifier relationships and profiles that the data statement is based on, expressed as references to those and their formulae, the date and time of release, the subject of the released data, the protocol transcript of the data exchange protocol, annotations by the party (user), the session identifier, and further metadata required for bookkeeping.

Relating a formula in the data track to the data handling policy it has been released under allows the party to later assess in an interaction with the other

---

[12] This case is prominent for a service provider who releases data about itself to anonymous uses who have not (yet) provided data about themselves. For service providers it is less interesting to track to whom data about themselves has been released to.

party the enforcement state of the data handling policy and, in case of issues, take further actions.[13] Annotations by the user may comprise comments, e.g., to distinguish multiple partial identities she has with the other party.

9.4.4   Profile Data

The profile data β of a party comprises a set of tuples, each comprising a profile record and a set of metadata predicates $(\beta_i, \{m_{\beta_i,j}\})$. A profile record β_i is a set of tuples $(\beta_k^*, \{m_{\beta_k^*,l}\})$ each comprising a formula and a set of metadata predicates. Note that this structure is the same as the structure of the other classes of data. The subject identifier for the party a record β_i applies to is associated with the record through a metadata predicate.

Whenever a party obtains data about another party, e.g., by receiving them from the other party, from third parties, or through any other means, those data are stored in the profile record about the other party, thereby creating an identity profile of the other party. A user creates a profile for each service provider she interacts with to store the information she has obtained about the service provider. This is at least the information stored in the service provider's public key certificate, such as its distinguished name, country, and URL. During a negotiation, the user may obtain further data, e.g., about the reputation, assurance mechanisms, or trust assessment of the service provider. A service provider creates a profile for each customer it interacts with in order to perform its business processes. When a customer uses a transaction pseudonym, each time a new profile is created about them. This approach reflects exactly the proper use of pseudonyms by a party. Profile data needs to be handled according to the data handling policies agreed with the provider of the data. Following European data protection legislation, users do not need to enforce any policy on identity data received from a service provider, the case of data received from (about) other users is different and puts users into the role of data controllers.

9.4.4.1   Data Formula

The identity data is represented through formulae expressed in our data model. Valid formulae are those that comply with the syntax of the data representation language. We do not give details here as the formulae are similar to the ones for representing the other kinds of data, only with possibly different expressivity.

The other party having released such a formula may have renamed terms in the formula before sending it in order to not establish linkability with other transactions by the other party or about the same subject. In an interpretation of the formula, the terms still refer to the same objects, as usual.

---

[13] One can think of an electronic assistant for filing a complaint with the data protection authority responsible for the data processing.

9.4.4.2   Metadata

One metadata predicate on a profile record is the subject term of the party the profile is about. All formulae received from or about the other party under the same subject term can be stored under the same profile record.

For a formula, the following are examples of metadata items to be stored: date and time of reception of the data, party identifier whom the data have been received from, protocol transcript of the data release protocol including all cryptographic tokens, and other metadata required for bookkeeping.

9.4.4.3   Use

The profile data has multiple uses in our architecture. When evaluating an access request to a resource of the party, profile data of the other party may be used to supply data for the authorization decision in addition to the dynamic subject profile. This is possible only in case of pseudonymous or identified interactions. When a user browses her data track, parts of the profile data of the data recipients can be mapped into the view the user is presented with in order to provide an enhanced user experience when assessing her releases of data. For a user's access to data, her profile about the other party specifies relevant information on how to access data at the other party, for example, the URL under which this service is provided.

9.4.5   Data Statements and Requests

When a party releases a data request or a data statement to another party, the data model is used to express such. A data request is formed as explained in Section 9.3 as formula with free variables and can be answered with one of possibly multiple valid responses. A data statement made to another party is either done in response to a request or proactively, in anticipation of a future request.

A data statement can make full use of the expressiveness of our data model. It is important to note that the expressiveness that can be used by a party for a particular data statement is constrained by the protocol it uses to proof the statement correct. For the user-centric model of data exchange, private certificate systems have substantial expressive power while allowing for privacy-enhanced operation. Over the wire, a data request or statement can be accompanied by a set of metadata predicates on the data (request) formula in order to express additional information not being captured by the data model. Metadata predicates can apply to specific items in the formula, e.g., an identity or identifier object. Such metadata can, for example, indicate whether a discjuntion in a request is allowed to be fulfilled by proving it as a single statement or only through fulfilling one of the sub-formulae comprising the disjunction where the latter is less data minimizing.

9.4.5.1   Summary

Recapitulating the classes of data held by a party as outlined above as well as data on the wire, that is, data statements and requests, we have defined the data representation following a common pattern: For each class of data, the party may store a plurality of entries, each of which relates to at least one data formula with associated metadata. Metadata can also be associated directly with entries. Re-using the same concepts for all kinds of data allows us to use the same technical means for representing and processing the related data and thus simplifies the architecture from both a technical as well as a conceptual standpoint. Along those lines, it would allow, from a conceptual point of view, to model all data in one repository and assign each entry to the class it belongs to by associating appropriate metadata—essentially a tag— with it. Though, this would harm the advantage of the current approach in terms of presentation to the reader.

Data of all kinds should be subject to a data life-cycle management for automatically enforcing the agreed or party-chosen data handling policies on specific data items or types of data. This is particularly true for profile data as those constitute the major part of user data stored by parties in standard scenarios. Examples for actions related to data life-cycle management are time-driven deletion of customer data, encrypted archival of non-active customer data, or user-notification on releases of the data to third parties or security breaches.

9.5   Identity Management Concepts

We next discuss some of the well-known concepts in identity management that are relevant for the design of our architecture. Furthermore, we extend or redefine the meaning of the terms where appropriate to integrate with additional features of our architecture.

9.5.1   Partial Identities

A *partial identity* has originally been defined as a set of identity-related attributes of a person. A partial identity thus exposes a certain facet of the identity [14] of a person and the person should be in control of defining and using partial identities at her own discretion. That is, a person can decide on her own which partial identity to expose to another party in an interaction.[PH10]

As we take the approach of data minimization even further than envisioned in previous work on the subject, we also generalize the definition of the concept

---

[14] The term identity is used here as the totality of attributes of a party, in contrast to our meaning of identity of being a named set of tuples each representing information on an attribute of the identity.

of partial identity of a subject to account for stronger ideas in terms of data minimization that we employ in this work:

Definition (Partial identity): A *partial identity* of a party is a set of formulae of our data model where the subject of the core identities referred to in the formulae refers to the party. The certification and other in-formula metadata are part of the partial identity as they comprise information relevant for assessing the trustworthiness of the attributes, such as who vouches for the data.   □

Like in the original definition, a partial identity exposes a certain facet of the identity of a subject (party) with the party having control over the shaping of her partial identities. The extension accounts for our requirement of expressing predicates over attributes as well as disjunctions between formulae to take data minimization even further without leaving the domain of efficient cryptographic protocols for releasing such data minimizing statements. Using such features allows a user to define a partial identity based on any such statement thus preventing certain unnecessary releases of attribute values where this is actually not required. Our changes retain the conceptual ideas behind the concept of partial identity and can be seen as an extension of technical nature in order to obtain better data minimization by using formulae talking about identities to specify the data about a subject instead of only sets of attributes. The concept of conditionally-released identities also contributes to the concept of partial identity in the sense that it comprises information that a data recipient may obtain under certain conditions. Both conditionally-released and opaque identities may, if used within a formula, reveal, through their relations to other identities, information on which third-party endorsed attributes the party has in its identity relationships.

Note that the concept of identity that we use deviates from the well-established term of identity as being the set of all identity attributes of a person. An identity in our definition is rather a set of triples of attribute name, operator, and value, certified by a certifier. Any entity can be subject of an identity, where entities can, among others, be natural persons or legal persons.

One user-side functionality in our identity management architecture is the management of partial identities of the user. When a user interacts with other entities, she makes decisions on the release of data to those entities and thereby establishes partial identities with those entities. Concretely, the partial identity established with a party is comprised by the conjunction of all the formulae that are revealed to the other party during one or more interactions under the same pseudonym. The partial identity thus forms the complete knowledge, as modeled by the identity management system, of the other party regarding the party. Note that this definition does not take into consideration information obtained from other sources or the application-layer interaction which may allow the other party to link the information it has obtained from the same user under different pseudonyms.

A partial identity is always created implicitly when a user engages in inter-actions with an other party through the release of data to this party. Information about the partial identities of a user is available through her data track in the form of the individual releases of data to other parties: Each release to any other party is recorded in the data track and kept for future use by the party. The partial identity can be compiled from the individual release records at any time.

When the user has multiple partial identities with one other party, i.e., data released under multiple different pseudonyms, the user should receive support for selecting the appropriate pseudonym (partial identity) in each new inter-action. Also, this should comprise functionality for supporting the user in the decision on which data to release in an interaction based on already-released data and released partial identities. This functionality concretely supports the user in her choice of the data to reveal to the other party in order to achieve certain privacy properties, e.g., the largest-possible anonymity set the user resides in.

9.6   Data Exchange Architecture

A key functionality of our architecture is the privacy-enhancing exchange of certified data between parties. This comprises, as basic functionality, on the one hand the establishment of identity relationships of a party that define which party vouches for which data of the party, and on the other hand the use of such identity relationships for revealing the certified data of identity relationships to data recipients. Additional functionality includes the revocation of identity relationships and functionality related to the concept of identity escrow to achieve, among others, a trade-off between anonymity and conditional identifiability of parties. The qualifier "privacy-enhancing" of the data exchange protocol refers to the property that only the data as specified by a party is released in an interaction. This particularly means that undesired linkability with previous transactions and the excessive release of attribute information in addition to what is intended to be released by the party are minimized. Our architecture allows both for advanced privacy-enhancing protocols such as private certificate systems as well as traditional protocols to be used. Only the further are capable of combining the mentioned privacy-enhancing properties with strong accountability features in a strong trust model. Our focus is, as this work deals with privacy-enhancing identity management, clearly on the privacy-enhancing protocols. From a perspective of architectural components, the data exchange functionality is encapsulated within the *data exchange component* of the architecture.

We stress that data exchange is concerned only with the mechanisms and protocols for exchanging well-specified data statements among parties and that it is not concerned with any "use" of the data by the recipient, e.g., for authorization or decisions on which data to request. Utilizing the obtained

data and deciding on which data to request from other parties is left to other components of the architecture, such as the authorization or the negotiation components. Thus, the data exchange component can be seen as one that implements the mechanics of data exchange. It is merely a tool to facilitate other functions of the architecture that operate on the data.

The architecture for the exchange of data has been designed such that different protocols for attribute exchange can be integrated into a unified framework that offers a single API towards the outside of the component. This API allows for declaratively specifying the data requirements for each transaction and is largely technology independent. This makes it possible, for example, to use as schemes for private certificates both the schemes by Camenisch and Lysyanskaya [CL03, CL04] and the one of Brands [Bra00] as privacy-enhancing data release protocols. The protocol implementations are encapsulated in sub-components of the data release component.[15] We refer to a sub-component implementing a protocol as *protocol engine*. The approach of abstracting the interface and encapsulating the protocol engines has the goal of allowing for the integration of different schemes into our architecture without (major) changes to available software components implementing the schemes, which can be complex components on their own, as is the case for the Identity Mixer component.

The tight integration of the interface of the component with our data model allows for leveraging the concepts of opaque identities as well as conditionally-released identities in addition to "standard" identities, where only the latter represent immediately-obtainable attribute data when being used in a data exchange protocol. See Section 9.3 for the introduction of those concepts. The support of those concepts advances the functionality of the component towards better privacy as well as accountability at the same time, that is, those properties are not compromising each other any more when using private certificate systems as data exchange protocols.

In the remainder of this section we present the architectural ideas for the data exchange component. We discuss its internal high-level architecture, the components it comprises, and the details of its protocol-independent interface. We outline, mainly in the explanation of the interface, how the Identity Mixer private certificate system can be integrated into the component as one specific protocol suite for data exchange with a focus on privacy-enhancing properties.

---

[15] Note that the componentization of the architecture is conceptual and a component being a sub-component does not imply containment in terms of implementation concepts or residing on the same physical machine. A sub-component can, for example, be implemented as a different sub-system residing on a separate machine or set of machines, and communicating with its super-component following the constraints imposed by the architecture in terms of its interface.

9.6.1   Roles in an Attribute Exchange Scenario

In a scenario for (privacy-enhancing) attribute exchange parties act under the *roles* as outlined below. A party can take on multiple roles at different times in different interactions or in a single interaction. Any party can act under any of the roles from an architectural perspective. Concrete use cases may constrain which party can take on which roles. Depending on the underlying protocol being used, it is possible that certain roles collapse into a single party. We show the relation between parties and the roles they act under in Figure 9.3.

Certifier: A certifier is a party that *vouches for data* about a data subject towards data recipients. This means, it certifies a data statement (formula) about the subject. When using appropriate protocols, parts of the data statement can later be revealed to data recipients with the guarantee that the data are those vouched for by the certifier.

Data provider: A data provider *establishes identity relationships* and *releases data* based on identity relationships to data recipients.[16] More concretely, the data provider is the party making decisions to do so and initiating the related protocols. Third parties may be involved in the mechanics of carrying out the protocols. The data provider is in many cases the subject of the data, which is particularly true in user-centric data release using private certificate systems. A notable exception to this are user-centric delegation scenarios when a data provider can, as delegatee, release data about a different party who is the subject of the data.

Data recipient: A data recipient is a party who *receives data statements* about subjects, vouched for by certifiers and made by data providers. A data recipient can be ensured that, if the protocol has been properly executed, the obtained data represent what the certifiers have vouched for, as specified in the certificiation metadata parts of the obtained statement.

Conditional data recipient: A conditional data recipient is responsible for *realizing the conditional release functionality* of releasing data. It is trusted by the data subject to not obtain the attributes of the conditionally-released identity unless an agreed condition is fulfilled and it is trusted by the data recipient or third parties, such as law enforcement agencies, to do so when the condition is fulfilled. In the Identity Mixer private certificate system this role can be acted under by a distinct party, thus enabling a strong model with separation of concerns.

We note that the subject or data subject is not a role itself as it only addresses whom data are about, but is not a role in the sense that a party can perform actions or be involved in protocols under the role. As in the user-centric model of identity management that we focus on, a data provider is often the subject

---

[16] The name "data provider" does not reflect the functionality of establishing identity or identifier relationships, though we do not introduce another role only for naming reasons.

of the data itself, we sometimes also refer to the data provider as subject if the meaning is clear from the context.

Figure 9.3 shows a single configuration of parties in terms of roles they act under. This can change during the same interaction (session), then, for example, the parties acting in the roles of data provider and data recipient swapping their roles. The latter is the case in a typical instance of a negotiation protocol two parties engage in for mutually exchanging data.
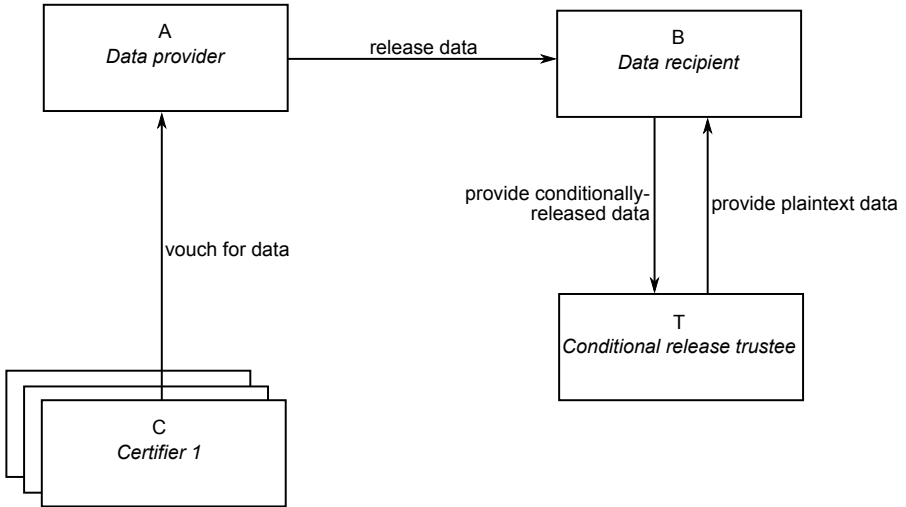
Fig. 9.3 Roles for the privacy-enhancing exchange of data

9.6.2   Private Certificate Systems

We next explain *private certificate systems* and briefly discuss their properties because private certificate systems are the main foundation in terms of protocols of our work on the component for privacy-enhancing data exchange.

A *private certificate system* is a system for the exchange of attribute data, certified by a certifier, between a data provider (possibly the subject) and a data recipient. It is based on a special kind of certificates, so called *private certificates* which are obtained by the data provider and later used to release parts of the attribute information contained in the certificates to data recipients. A private certificate system ensures attribute integrity, that is, only attribute data certified by a certifier can be revealed in a transaction by the user to a data recipient. Private certificate systems have the following privacy properties: Selected parts of the attribute information of a certificate can be released in an interaction, such as individual attributes or predicates

on attributes; multiple uses of the same certificate do not make the resulting transactions linkable, unless the released attribute information makes the transactions linkable (multi-show unlinkability); a balance between privacy (anonymity) and accountability can be realized.

A private certificate endorses, considering its technical realization, a tuple $(a_1, \ldots, a_k)$ of attributes with the value domain being a large integer interval defined by the parameters of the system. The private certificate comprises a signature σ on the tuple of integer attributes, where preferred schemes are the SRSA-CL scheme [CL03] and the BL-CL scheme [CL04] for their strong properties in terms of achieving multi-show unlinkability.

In addition to private certificates, such systems can handle pseudonyms, that is, identifiers of parties held with other parties under which certificates can be obtained and information therein released.

In order to inject attribute semantics into a private certificate which is merely a cryptographic entity, a type and a certificate structure are used. A type, corresponding to the identity type, specifies the ontology types, data types, and other relevant aspects of the identity the private certificate is related to and thereby provides a link to the semantics of our data model. Each certificate type has exactly one certificate structure associated that defines a mapping from the data semantics to the technical realization of the certificate. See Section 9.6.6.1 for details on the certificate structure and how it is realized.

In our architecture, a private certificate is specified precisely by its corresponding identity relationship, the formulae of which describe the certificate consistent with the data the certificate comprises as explained in Section 9.4.

In the literature, a diverse nomenclature exists for the concept of private certificates and closely-related concepts (and the systems realized with them): anonymous credentials, private certificates, private credentials, or minimal disclosure tokens. Anonymous credentials can be considered a term covering all such systems. We chose the name private certificates in order to refer to systems that are based on a blinding of the certificate in the proof protocol and thus have the capability of multi-show unlinkability such as the Identity Mixer system. Anonymous credential systems do not necessarily have this property.

9.6.3   High-Level Architecture

In this subsection we discuss the high-level architecture of the data exchange component. Figure 9.4 presents its component architecture. The component has an interface agnostic to the protocol being used for data exchange, exposed to the invoking component. A *protocol engine* implements all protocols, or a relevant subset matching the roles the party takes on, of a protocol suite, and is thus a main kind of sub-component. The protocol engines implement the core of the functionality of the data exchange protocols and multiple protocol engines can be added to an instance of the data release component.

The protocol engine we focus on in this section is the one implementing the Identity Mixer credential system protocol suite. Each protocol engine is connected to a corresponding *protocol adaptor*. A protocol adaptor is responsible for integrating a protocol engine with the data exchange component, that is, bridging the gap between the possibly proprietary interface of the protocol engine and the unified interface exposed by the data exchange component. The adaptor can furthermore connect to other (protocol-specific) components needed for executing the protocols of the protocol engine or keeping state between protocol runs, e.g., a key management component for handling required cryptographic keys or a token storage component for storing (cryptographic) tokens. Those additional components can be dedicated to a specific protocol engine or shared between multiple engines. A key management component for handling public keys of other parties is a good example for a shared component used by multiple protocols. A *data release orchestrator* connects the API of the data exchange component with the APIs of each of the protocol adaptors. This orchestrator is responsible for interacting with (a subset of) the protocol adaptors for a given input and processing inputs and outputs of the sub-components it is connected with.

The figure shows the data exchange component instantiated with a protocol engine for the Identity Mixer private certificate system, the corresponding protocol adaptor and dedicated token storage component, and a shared key management component. It shows also a second protocol engine with its protocol adaptor that uses the shared key management component. The design idea behind allowing additional components to be used by one or more of the protocol engines is the flexibility of this approach of having the core integration work for a protocol engine done by its protocol-specific protocol adapter and further protocol-specific or protocol-independent functionality by the additional components. This improves the capabilities of the component's architecture of integrating with available protocol engine implementations. The sub-components of the data exchange component can access outside components, such as data repositories, or establish or use communication channels with other parties. We discuss further below in this section the integration of the Identity Mixer private certificate system into the data exchange component as the main data exchange protocol we build on within this book.

9.6.4 Component Interface

The data exchange component has a single unified interface towards the other components of the architecture, regardless of the data exchange protocols being supported. This interface is based, to a large extent, on our formalism of expressing data – our data model. In this subsection, we present the details of the interface and particularly discuss it in the light of using private certificate systems as underlying data exchange mechanisms. The interface exposed by the component is invoked from other components of the architecture, such as the negotiation component that drives mutual attribute exchange between

API towards other components
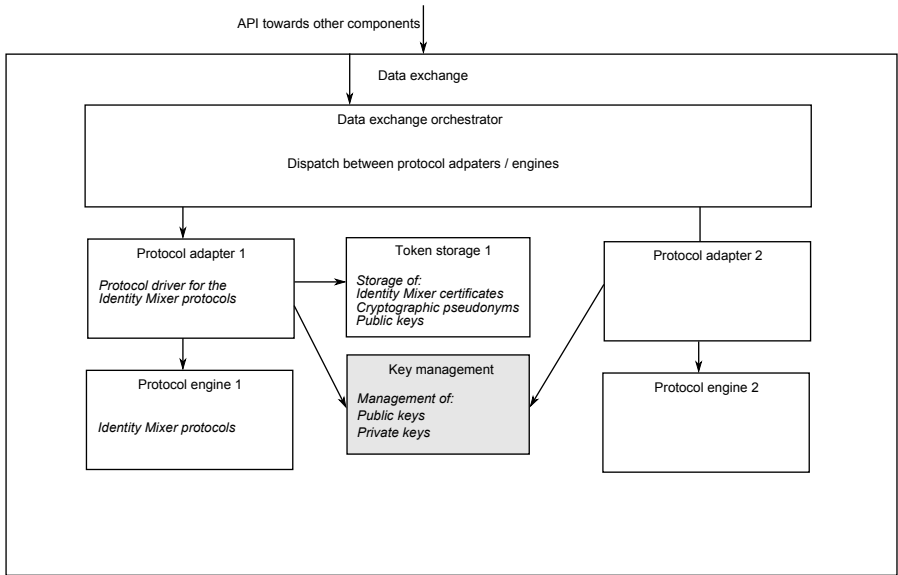
Fig. 9.4 The data exchange component

parties (see Section 9.9). From a perspective of modeling, the API has been crafted in a way such that it allows for the declarative use of the component based on identity concepts, without referring to or being restricted by particular underlying technologies of specific data exchange protocols and the concrete protocol flows. Particularly, the processing and handling of cryptographic material is fully encapsulated into the component and outside components need not implement protocol-specific extensions for different protocols. The main purpose of the protocol interface is to be an abstract API that decouples the data release component from the rest of the architecture. Other components do not require adaptation of their code or API when a new protocol is added to the data release component.

The protocol-independent way of modeling the interfaces of the component is achieved by specifying *abstract protocol interfaces* for the protocols that implement the functionality of data exchange. Those protocol interfaces specify the input and output (I/O) requirements of the protocols for data exchange regardless of the technology used. Concrete protocol suits provide concrete implementations of those interfaces within their protocol engine subcomponents. The underlying message flow for a protocol that will be executed in response to an invocation at the abstract level of the protocol interface is protocol dependent and may, as noted next, involve additional third parties required for the mechanics of the protocol.

Each protocol interface is specified only for the two parties which are the conceptual endpoints of the protocol in terms of identity management,

leaving out possibly involved third parties. If in a specific protocol a third party needs to be involved, the interface of this third party needs to be defined in addition for integrating the protocol. It is impossible to define those in a generic way as the flows may differ considerably depending on the protocol suite considered. The third parties can be seen as instruments for executing certain data exchange protocols requiring involvement of such third parties.

Let us next make some general observations concerning all protocol interfaces. A general rule for the interface is that the return values may supersede the input values and thus may need to be stored with the corresponding identifier relationship or identity relationship that is being created. Concretely, for those protocols, those returned formulae have all free variables instantiated and specify, in contrast to the input, a concrete identifier relationship or identity relationship.

For all protocols, metadata can be provided as input and received as output at a protocol endpoint. We model the interface in a general way with respect to metadata by allowing for a set of metadata predicates to be associated with the formula itself, or objects referred to in the formula, by labelling a set of metadata predicates with the object it is to be associated with. For the avoidance of doubt, we stress that the metadata are mostly used as inputs for the cryptographic protocols to be executed and sometimes, but not necessarily, transferred to the other party. Identifiers that would establish undesired linkability are of course never exposed to the other party.

The interfaces we present may hide certain elements, concretely, data that is read/written from/to the party's data repository by the component. Thus, the interface does not reflect everything the protocol needs or generates, but is designed in a way to be as simple to use by other components as possible and abstracting from complexity whereever possible. This approach leads to a strong coupling of the component with the data repositories which we find a useful trade-off to be taken. This may be given up for the sake of reducing the coupling while complicating the interface.

We next present the interfaces for the protocols for establishing a subject identifier, for establishing an identity relationship, and for the release of data in detail and discuss them with respect to using the Identity Mixer system as concrete protocol suite for data exchange.

9.6.4.1   EstablishSubjectIdentifier

The protocol *EstablishSubjectIdentifier* is used by a party $\mathcal{A}$ to create a new identifier as a reference to itself towards party $\mathcal{B}$. The resulting identifier is, if backed by a cryptographic token for proving its holdership, a pseudonym following the definition in [PH10] with the slight deviation that we do not restrict the identifier value from comprising further attribute semantics than being a semanticless identifier. This new identifier can be used by $\mathcal{A}$ to talk about itself with $\mathcal{B}$.

Depending on the underlying protocol that implements the functionality, further parties besides $\mathcal{A}$ and $\mathcal{B}$ can be involved in the message flow of the actual protocol being executed. In the case of establishing a pseudonym with the Identity Mixer protocol suite, the involved parties are $\mathcal{A}$ and $\mathcal{B}$ between which the pseudonym is established and no further parties, thereby preventing other parties from obtaining information on the identifiers a party establishes with other parties. In the case of a protocol involving a party of the kind of a certifier, this party may be involved as well, but is not made explicit in the protocol interface.[17]

We next present the inputs and outputs of the parties $\mathcal{A}$ and $\mathcal{B}$ for executing the protocol where $\phi_{\mathcal{A}}$ and $s_{\mathcal{B}}$ are the inputs fully specifying the intentions for the protocol instance to be executed.

EstablishSubjectIdentifier

Input	
$\mathcal{A}$	$\phi_{\mathcal{A}}, s_{\mathcal{B}}, \{(o_i, \{\nu_{l_{\mathcal{A},i}}\}_{1 \le l_{\mathcal{A},i} \le n_{\mathcal{A},i}})\}_{1 \le i \le r_{\mathcal{A}}}$
$\mathcal{B}$	$-$
Output	
$\mathcal{A}$	$\phi'_{\mathcal{A}}, \{(o'_j, \{\nu'_{l'_{\mathcal{A},j}}\}_{1 \le l'_{\mathcal{A},j} \le n'_{\mathcal{A},j}})\}_{1 \le j \le r'_{\mathcal{A}}}, success_{\mathcal{A}}$
$\mathcal{B}$	$\phi'_{\mathcal{B}}, \{(o'_k, \{\nu'_{l'_{\mathcal{B},k}}\}_{1 \le l'_{\mathcal{B},k} \le n'_{\mathcal{B},k}})\}_{1 \le k \le r'_{\mathcal{B}}}, success_{\mathcal{B}}$

Using the syntax of our data model for representing identifier relationships leads to the following formulae, $\phi_{\mathcal{A}}$ for the input of party $\mathcal{A}$, $\phi'_{\mathcal{A}}$ for $\mathcal{A}$'s output, and $\phi'_{\mathcal{B}}$ for party $\mathcal{B}$'s output. The free variables in the input $\phi_{\mathcal{A}}$ have the meaning that they will be instantiated through the protocol. We note that $\phi_{\mathcal{A}}$ can be derived from a template at party $\mathcal{A}$, similar to what is done for identity relationships. We give an example next for the inputs and outputs for a protocol for a party $\mathcal{A}$ establishing a new pseudonym with party $\mathcal{B}$ using the Identity Mixer private certificate system.

$$\phi_{\mathcal{A}} = Eq(\mathsf{P}.subject, \mathsf{user4}) \wedge Eq(\mathsf{P}.subjectId, \mathsf{SubjectId}) \wedge$$
$$Eq(\mathsf{P}.protocol, \mathsf{Identity\_Mixer\_2048\_bit})$$
$$\phi'_{\mathcal{A}} = Eq(\mathsf{p}.subject, \mathsf{user4}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge$$
$$Eq(\mathsf{P}.protocol, \mathsf{Identity\_Mixer\_2048\_bit})$$
$$\phi'_{\mathcal{B}} = Eq(\mathsf{p}.subject, \mathsf{user4567}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge$$
$$Eq(\mathsf{P}.protocol, \mathsf{Identity\_Mixer\_2048\_bit})$$

For the input $\phi_{\mathcal{A}}$, the identifier object can be represented through a free variable P for a new identifier object or a constant term for an existing identifier

---

[17] The reason for this approach is that we focus on user-centric protocols, that is, such where the user itself establishes the identifier relationship with another party, without involvement of another trusted party. This is aligned with the goal of reducing trust in third parties as much as possible through the use of cryptography for our protocols.

object in case that an identifier is to be "extended" towards another party (see further below), the subject of the identifier to be created is specified by the attribute *subject*, in the example above the constant user4. The new identifier to be established for $\mathcal{A}$ through this protocol execution can be specified through a variable for the case of a new identifier to be created—SubjectId in the example—or a constant for the case of an existing subject identifier being established towards a further party. The protocol being used for establishing the identifier relationship is indicated through the *protocol* attribute. Note that we may omit this attribute in our examples for reasons of brevity. Party $\mathcal{B}$ is specified through the subject term $s_{\mathcal{B}}$ it is referred to by the party $\mathcal{A}$.

The output of party $\mathcal{A}$ is a new formula $\phi'_{\mathcal{A}}$ with all free variables being instantiated with values created during the protocol execution. Concretely, the new subject identifier *subjectId* created for $\mathcal{A}$ with $\mathcal{B}$ is user4567. The *subject* term for referring to $\mathcal{A}$, that is, the term $\mathcal{A}$ uses for referring to itself in its local data representations, is user4. We adopt the convention of always using the same term by a party for referring to itself as otherwise functionality of the system in terms of, e.g., matching locally-held data with policies, would be impaired. Party $\mathcal{A}$ also receives the following metadata as output: the private cryptographic material of the identitifier relationship that has been established, being a private part of an Identity Mixer pseudonym in the case of using this protocol suite; the subject identifier $s_{\mathcal{B}}$ of $\mathcal{B}$ with whom the identifier has been established; the mapping from term user4 to user4567 to be used for itself as subject when communicating under this identifier to the other party; other metadata such as time of protocol execution for internal bookkeeping.

The output of party $\mathcal{B}$ is a new formula $\phi'_{\mathcal{B}}$ with free variables being instantiated with values created during the protocol execution, very similar to $\phi'_{\mathcal{A}}$ above. The difference is that the term referring to $\mathcal{A}$, the subject of the identifier (pseudonym), is a different term as the one in $\mathcal{A}$'s output. In the example above, the term user4567 is used. We adopt the convention that the term that is used to address the subject of an identifier object is the same string as the *subjectId* of the identifier, though, from a different value domain. The crucial difference between the two is that the *subjectId* attribute is always interpreted with itself while the *subject* is interpreted with the party it refers to, thereby enabling certain features of making deductions and finding fulfilling assignments for data requests. $\mathcal{B}$ receives metadata as output: the (semi-)public part of the cryptographic material representing the identifier relationship, being a public part of an Identity mixer pseudonym in case of using this protocol suite; and the usual metadata as above. $\mathcal{B}$ learns nothing more than its output data presented above, particularly not any further information about $\mathcal{A}$, which is a main property of a pseudonym protocol.

Regarding the renaming of terms in order to avoid unintended linkabilities between actions of the party, the rule is that whenever a mapping is defined in the metadata of a formula of an identifier relationship, the mapping must be used to rename locally-used terms in a formula before sending the formula

to the other party. The application of such a mapping is required to maintain unlinkability of the transaction to previous transactions.

Once the protocol has been successfully executed, entries in the data repositories are created by the parties as follows: $\mathcal{A}$ creates a new entry in its identifier relationships comprising the formula $\phi'_{\mathcal{A}}$ and the output metadata. It is flagged as one of the party's identifier relationships it has established with another party through a metadata predicate. To reflect the data exchange with $\mathcal{B}$, $\mathcal{A}$ creates an entry in its data track with itself being the subject, that is, subject term user4 being used, and recipient being $s_{\mathcal{B}}$ and data being formula $\phi'_{\mathcal{A}}$. $\mathcal{B}$ creates a corresponding identifier relationship entry with formula $\phi'_{\mathcal{B}}$ and subject user4567 flagged as the recipient side of it through metadata. $\mathcal{B}$ also creates an entry in its profile data for subject term user4567, with formula $\phi'_{\mathcal{B}}$.

Protocol Variants

The protocol interface explained above allows for establishing a single subject identifier with one other party, the standard case of pseudonyms in Identity Mixer. When considering the valid case of establishing the same pseudonym with multiple other parties (multi-party identifier) or the public, that is, all parties in the system, either an extension of the above protocol interface or a new protocol is required. The way this is modeled is an API design decision – we model it with the same protocol *EstablishSubjectIdentifier* that is parametrized accordingly with an already-existing pseudonym value in order to be able to use this pseudonym with further parties.

We do not give all the details, but rather sketch the use through an example, based on the previous example, for extending an already-established pseudonym to another party.

$$\phi_{\mathcal{A},\mathcal{B}_2} = \textit{Eq}(\mathsf{p}.subject, \mathsf{user4}) \wedge \textit{Eq}(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge \ldots$$

The example input formula $\phi_{\mathcal{A},\mathcal{B}_2}$ of party $\mathcal{A}$ and the additional input $s_{\mathcal{B}_2}$, an identifier $\mathcal{A}$ uses to refer to the party $\mathcal{B}_2$, specify that the same pseudonym identifier user4567 is to be established with $\mathcal{B}_2$ as well. New entries in $\mathcal{A}$'s identifier relationships, data track, and $\mathcal{B}$'s identifier relationships and profile data are created once the protocol has been successfully executed. The output formulae are analogous to the previous example. By executing multiple instances of this protocol and always using the same pseudonym identifier, party $\mathcal{A}$ establishes the same pseudonym identifier with further parties.

When a party $\mathcal{A}$ creates a *public pseudonym*, that is, a public name, it executes a single protocol, very similar to the *EstablishSubjectIdentifier* protocol, in which it creates the private and public formulae and cryptographic objects for the pseudonym and makes the public part available to all parties in the system. The latter can be achieved by standard means of uploading the public part to a public pseudonym repository. A special case of this are public keys of parties on the Internet as of today, this special case being one

of the motivations for introducing public pseudonyms as it allows us to model identifiers of service providers and certifiers in exactly the same way as the ones of users. Let us note that when using the above protocol, the pseudonym is created and uploaded to a repository, but no pseudonym has yet been established by $\mathcal{A}$ with any other party – this happens only later once another party obtains the public part of the pseudonym from the repository.

9.6.4.2 EstablishSubjectIdentifierO

This protocol is used by a party $\mathcal{A}$ to establish a subject identifier about a party $\mathcal{C}$ with party $\mathcal{B}$. The input is, similarly to all our protocol interfaces, based on the data representation of our data model. The name of the protocol is inspired by appending an "O" for "other party" to the name of the basic protocol. We briefly sketch the protocol interface by explaining its main differences to the protocol *EstablishSubjectIdentifier* presented above and do not present all details. Let us for this assume that $\mathcal{A}$ and $\mathcal{C}$ already have pseudonyms established with each other and $\mathcal{A}$ knows $\mathcal{C}$ by the *subjectId* user1234 and refers to subject $\mathcal{C}$ with subject user1234 locally, following our naming convention. In the protocol, $\mathcal{A}$ wants to establish a new subject identifier for $\mathcal{C}$ with party $\mathcal{B}$, where $\mathcal{A}$ is talking to $\mathcal{B}$ under the pseudonym identifier user4567 established in the example above. We give the input and output formulae next:

$$
\begin{aligned}
\phi_{\mathcal{A},2} =& Eq(\mathsf{P}.subject, \mathsf{user1234}) \wedge Eq(\mathsf{P}.subjectId, \mathsf{SubjectId}) \wedge \\
& Eq(\mathsf{p}.subject, \mathsf{user4}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge (\mathsf{P} \neq \mathsf{p}) \\
\phi'_{\mathcal{A},2} =& Eq(\mathsf{p}_2.subject, \mathsf{user1234}) \wedge Eq(\mathsf{p}_2.subjectId, \mathsf{user3456}) \wedge \\
& Eq(\mathsf{p}.subject, \mathsf{user4}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge (\mathsf{p} \neq \mathsf{p}_2) \\
\phi'_{\mathcal{B},2} =& Eq(\mathsf{p}_2.subject, \mathsf{user3456}) \wedge Eq(\mathsf{p}_2.subjectId, \mathsf{user3456}) \wedge \\
& Eq(\mathsf{p}.subject, \mathsf{user4567}) \wedge Eq(\mathsf{p}.subjectId, \mathsf{user4567}) \wedge (\mathsf{p} \neq \mathsf{p}_2)
\end{aligned}
$$

The protocol inputs of $\mathcal{A}$ are the formula $\phi_{\mathcal{A},2}$, the *subjectId* for $\mathcal{B}$, $s_{\mathcal{B}}$, and the *subjectId* user1234 indicating the subject for whom the pseudonym is to be created. The above formulae show the input and output formulae of the parties. Again, analogous comments regarding naming of terms apply as for the protocol interface *EstablishSubjectIdentifier*.

Once the protocol has been executed with success, $\mathcal{A}$ creates an identifier relationship entry with $\phi'_{\mathcal{A},2}$ and metadata for the subject identified through term user1234 used at $\mathcal{A}$ for referring to $\mathcal{C}$. This pseudonym entry, as it is related to the party referred to by subject user1234, that is, to some party, and not $\mathcal{A}$ itself, can later be used by $\mathcal{A}$ to refer to $\mathcal{C}$ towards $\mathcal{B}$. As the formula expresses the subject term for the subject, it can match with appropriate data requests asking for a pseudonym $\mathcal{A}$ holds for another party. It is worth noting that $\mathcal{A}$'s pseudonym is included in the formula in order to enable standard

matching processes of data with requests as specified in Section 9.3 on the data model.

$\mathcal{B}$ creates an identifier relationship entry for subject term user3456, the term used to refer to $\mathcal{C}$, comprising the formula $\phi'_{\mathcal{B},2}$, metadata, and data provider user4567. It also creates a profile data entry. When further communicating with $\mathcal{A}$ about $\mathcal{C}$, the subject term user3456 will subsequently be used by both parties to refer to $\mathcal{C}$ following our convention on term naming. Deviating from this would break certain capabilities of matching data with policies and making derivations, and other processes that rely on syntactic derivations over formulae.

Regarding the modeling of such an identifier relationship as above, we want to note that the predicate stating that the two identifier objects in the formula are not equal binds them together in a way that when in a request the two identifiers are required with the non-equals predicate between them, only this formula will match this request. Without this predicate, the parts could be combined from different formulae of the party to fulfull the policy which is not the intention. As an alternative we could use a new predicate *Linked* that explicitly performs a binding between the two identifier objects. A policy requesting an identifier statement about another party will typically also comprise a predicate to indicate which party is the communication counterpart in order to make the roles of the subjects in the formula clear. This is important in a delegation setting.

Besides this case of a party $\mathcal{A}$ establishing a pseudonym for a party $\mathcal{C}$ with $\mathcal{B}$, there is also the possibility that only a new identifier is created by $\mathcal{A}$ about $\mathcal{C}$ when talking about $\mathcal{C}$ with $\mathcal{B}$, without this being backed by a cryptographic pseudonym. Conceptually, this is analogous, though, the trust model behind it is quite different as honesty of party $\mathcal{A}$ is assumed. It can be realized also by party $\mathcal{A}$ obtaining a new entry in its identifier relationships. We do not give further details on this.

We note that the approach of this protocol of linking the identifier of $\mathcal{A}$ with $\mathcal{B}$ to the established identifier for $\mathcal{C}$ in a single identifier relationship is the natural approach in terms of identity management: Using the latter identifier with a different identifier of $\mathcal{A}$ with $\mathcal{B}$ would immediately link the identifiers of $\mathcal{A}$ with $\mathcal{B}$ under which the identifier for $\mathcal{C}$ with $\mathcal{B}$ is used, thus from a perspective of identity management it is not useful to use the identifier for $\mathcal{C}$ under different identifiers of $\mathcal{A}$ with $\mathcal{B}$.

9.6.4.3   EstablishIdentityRelationship

The protocol *EstablishIdentityRelationship* creates a new identity relationship between a user and a certifier. As explained in detail in Section 9.2.3, an identity relationship specifies an identity vouching relationship between two parties and can, once established, be used by its holder to reveal certified attribute data about the subject to other parties. Establishing an identity relationship means that the certifier agrees to certify the data related to the subject in the

identity relationship and enabling the holder of the identity relationship to utilize the identity relationship in future interactions to release parts of the certified attribute data contained therein. Using an identity relationship can be done without further involving the certifier when using user-centric protocols for releasing data, for example private certificate systems.

The challenges of designing a clean interface for this protocol are multifold: achieving high expressivity; the data formula passed as input may change due to the use of opaque identities; consideration of delegation; and the creation of the new identity relationship may be related to previous protocol exchanges with the party thus requiring access to and the capability of referencing (cryptographic) objects of those previous exchanges. We have found the strong integration with our data model to be crucial for obtaining a practically functional system. We next present the protocol interface from both the view of the certifier and the prospective holder (e.g., a user). Let party $\mathcal{A}$ be the prospective holder and party $\mathcal{B}$ the certifier.

The following is the protocol interface for the protocol *EstablishIdentityRelationship*, executed between a party $\mathcal{A}$ and a party $\mathcal{B}$, the latter being the certifier.

EstablishIdentityRelationship

$$
\begin{array}{ll}
\textbf{Input} & \\
\mathcal{A} & - \\
\mathcal{B} & \phi_{\mathcal{B}}, s_{\mathcal{A}}, \{(o_i, \{\nu_{l_{\mathcal{A},i}}\}_{1 \leq l_{\mathcal{A},i} \leq n_{\mathcal{A},i}})\}_{1 \leq i \leq r_{\mathcal{A}}} \\
\textbf{Output} & \\
\mathcal{A} & \phi'_{\mathcal{A}}, \{(o'_j, \{\nu'_{l'_{\mathcal{A},j}}\}_{1 \leq l'_{\mathcal{A},j} \leq n'_{\mathcal{A},j}})\}_{1 \leq j \leq r'_{\mathcal{A}}}, success_{\mathcal{A}} \\
\mathcal{B} & \phi'_{\mathcal{B}}, \{(o'_k, \{\nu'_{l'_{\mathcal{B},k}}\}_{1 \leq l'_{\mathcal{B},k} \leq n'_{\mathcal{B},k}})\}_{1 \leq k \leq r'_{\mathcal{B}}}, success_{\mathcal{B}}
\end{array}
$$

The *input* is explained first. The protocol input is determined by the certifier $\mathcal{B}$ as it is the party that decides on the data to be included in the identity relationship. Clearly, this data can be a function of the data received in the interaction or previous interactions with $\mathcal{A}$ or data obtained by other means by $\mathcal{B}$. A does not provide input to the protocol. The input of $\mathcal{B}$ comprises a data statement $\phi_{\mathcal{B}}$, $s_{\mathcal{A}}$, and metadata predicates expressed on the formula or identifiers and identities in it. The input $s_{\mathcal{A}}$ of $\mathcal{B}$ is the identifier value $\mathcal{A}$ is known to $\mathcal{B}$ under. This identifier must always be known by $\mathcal{B}$ as it needs to know the other party by some means, e.g., for purposes of allowing for anonymity revocation of $\mathcal{A}$'s anonymity as well as for certain protocols where it needs to be involved online when $\mathcal{A}$ uses the identity relationship to be established. We note that most metadata required by the protocol need not be passed at the API level, but can be obtained by the protocol adapter or engine from other components of the architecture or remote services, which leads to a further simplification of the interface. The output of $\mathcal{A}$ comprises a data statement $\phi'_{\mathcal{A}}$ as well as a set of metadata predicates returned at the side of $\mathcal{A}$; $\mathcal{B}$'s output comprises, analogously, a data statement $\phi'_{\mathcal{B}}$ and metadata predicates.

We next give details on the parameters of the API and particularly show the differences of the output to the input as there are some important aspects to note when using private certificate systems. The parameter that hides most of the interface complexity is the data parameter $\phi_\_$ at both of the parties.

The formula $\phi_\mathcal{B}$ and its corresponding output pendants are used to express the information as explained next. The formulae are formulae of our data model of Section 9.3. We refer the reader to Section 9.4 for the details on how identity relationships are expressed based on our data model.

Identity data: First and foremost, $\phi_\_$ precisely specifies the attribute data of the identity relationship to be created, expressed in our data model. This particularly includes ontology types ("attributes"), values or references to values of opaque identities, operators, and data types, thus forming the core of the data formula. For an identity relationship to be implemented through an Identity Mixer certificate, a single identity in the formula represents the attributes of the certificate. As one particular attribute of the identity, the subject the identity relationship is to be established for is specified through a subject term.

In-formula metadata: In-formula metadata comprises metadata expressed in the formula, that is, within our data model. The *certifier specification* is the first important kind of such metadata. The certifier of the identity relationship is specified as usual in the data formula $\phi_\_$ through a separate identity that is related to the identity representing the attribute data of the identity relationship through the *certifier* attribute. For protocols that require public and private keys, such as Identity Mixer, the certifier specification is required such that the protocol adaptor can obtain appropriate keys regarding the certifier. Keys are determined through the certifier and the identity type of the identity relationship to be established, e.g., an identity relationship of type Swiss_eID_card issued by the Swiss Government determines a key uniquely. The *temporal validity* of the identity relationship needs to be specified in the formula through values for the *validfrom* and *validuntil* attributes of the identity. This is part of the formula as it is important to be able to express statements on the temporal validity in a policy, e.g., when checking that an electronic passport is still valid for the whole duration of the intended stay when a person enters a country. In addition to the above, further in-formula metadata can be expressed if needed for certain data exchange protocols. A decision to move metadata that are otherwise not represented in the formula into the formula is a design choice, it allows for, e.g., being able to express authorization policies and do syntactical derivations over those data.

The following *on-formula metadata* can be expressed on a formula $\phi_\_$ through metadata predicates, that is, outside of our data model; thus, such on-formula metadata do not have meaning in the data model and authorization policies cannot build on such data and they are not considered in syntactical

derivations in our data model. Such metadata is input by $\mathcal{B}$ and output to both $\mathcal{A}$ and $\mathcal{B}$.

Technical specification: The technical specification of an identity relationship comprises any data that are required by the underlying protocols in addition to what is already modeled within $\phi_{\_}$ in the data model as described above. For Identity Mixer, an important part of this technical specification is the *certificate structure* as outlined in Section 9.6.6.1. The certificate structure is input as metadata on the identity variable in the formula. In a nutshell, a certificate structure is defined for each certificate type by the certifier for certificates of this type or a designated party and determines technical encoding details for the encoding of the data into cryptographic tokens. We note that this encoding does not have any meaning at the level of the data model, but rather is necessary for its technical implementation. This includes a mapping of attributes specified in the data formula to integer attributes supported by the cryptographic system and a specification of supported features of the certificate. The latter specifies what kinds of features the underlying technology of the identity relationship supports. For a relationship based on the Identity Mixer private certificate system, this can include the following for a certificate: k-show per time interval; one-show; supported credential revocation methods. We want to note that the technical information for an identity relationship is metadata that is public and the same for all identity relationships of a given type by a certifier.

Further on-formula metadata: Other on-formula metadata can be expressed in metadata predicates on $\phi_{\_}$. This may, for example, comprise cryptographic materials or references to such.

The formula used as input for the protocol for establishing an identity relationship needs to be created before the protocol execution. In our architecture, the issuer can use a *template* that acts as a blueprint for each new identity relationship. See Section 9.6.6.2 below for details on the template concept. We note that this is not part of the data exchange component, but rather done at a higher level.

The inputs discussed above are—in our view—sufficient for the requirements of today's privacy-enhancing attribute exchange protocols such as private certificate systems and have been derived mainly based on the requirements for exactly such kinds of systems. Other, less powerful systems in terms of privacy protection, have less stringent requirements on their protocol interface, e.g., in terms of data representation as well as supported features, and thus are likely to be subsumed by our approach.

We next give an example for establishing an identity relationship, continuing the examples above on establishing identifier relationships.

Example (Establishing an identity relationship (Input)): For the example, let us assume that party $\mathcal{A}$ has an opaque identity oid5678 with an attribute

ssn with value 6789987667899876 and that this opaque identity has been referred to in an interaction with $\mathcal{B}$ previously. That is, the opaque identity can be referred to by both parties $\mathcal{A}$ and $\mathcal{B}$, but $\mathcal{B}$ does not learn its attribute values. Both parties have related cryptographic tokens for performing proofs over the opaque identity.[18]

$$\phi = \textit{Eq}(\mathsf{Id}.subject, \mathsf{user4567}) \wedge \textit{Eq}(\mathsf{Id}.firstname, \mathsf{Jane}) \wedge \ldots \wedge$$
$$\textit{Eq}(\mathsf{Id}.ssn, \mathsf{oid5678}.ssn) \wedge$$
$$\textit{Eq}(\mathsf{Id}.validfrom, \mathsf{2009\text{-}07\text{-}01}) \wedge \textit{Eq}(\mathsf{Id}.validuntil, \mathsf{2009\text{-}12\text{-}31})$$
$$\textit{Eq}(\mathsf{Id}.certifier, \mathsf{germanGovernment}) \wedge \textit{Eq}(\mathsf{cid7560}.subject, \mathsf{germanGovernment}) \wedge$$
$$\textit{Eq}(\mathsf{cid7560}.uniqueid, \mathsf{german\_Government\_eID\_Issuer}) \wedge$$
$$\textit{Eq}(\mathsf{id4890}.protocolsuite, \mathsf{Identity\_Mixer\_2048\_bit})$$

The formula ϕ specifies the input for the protocol by the certifier $\mathcal{B}$. The identity variable Id is the term referring to the identity that is the basis of the to-be-created identity relationship. For representing the identity data and metadata, as usual, ontology types and attributes are expressed through relating the ontology types of the identity with the constant values by using predicates, e.g., for expressing equality. Other predicates such as inequalities may be used as well, depending on the underlying protocol. The *subject* is specified to be equal to the term $\mathsf{user4546}$ used by $\mathcal{B}$ for referring to the user. The part of the formula $\textit{Eq}(\mathsf{Id}.ssn, \mathsf{oid5678}.ssn)$ relates the attribute *ssn* of the new identity with the hidden attribute *ssn* of the opaque identity $\mathsf{oid5678}$. That is, the attribute $\mathsf{id4890}.ssn$ will receive the value $\mathsf{6789987667899876}$ in $\phi'_{\mathcal{A}}$ without $\mathcal{B}$ learning it. $\mathcal{B}$ only gets to know that the attribute *ssn* in the new identity to be issued has the same value as the *ssn* attribute of opaque identity $\mathsf{oid5678}$. The temporal validity of the identity relationship is specified to be from $\mathsf{2009\text{-}07\text{-}01}$ to $\mathsf{2009\text{-}12\text{-}31}$ through appropriate predicates. The identity $\mathsf{cid7560}$ is declared to be an identity relating to the certifier of the new identity by using the subject term $\mathsf{germanGovernment}$. We assume that this identity is a public one, thus known under the same name to possibly every party in the system. Also it is assumed that the same subject term to refer to the certifier is used by every party as there is no need for anyone to rename it. Its attribute *uniqueid* is specified to be equal to $\mathsf{german\_Government\_eID\_Issuer}$ which unambiguously defines the certifier in order to allow for the retrieval of the proper keys. The attribute *protocolsuite* specifies that the protocol suite to be used for realizing the identity relationship is to be $\mathsf{Identity\_Mixer\_2048\_bit}$. This determines the protocol suite that needs to be used whenever using the identity relationship to release data.   □

The *output* of the protocol comprises a data statement ϕ' on each side as well as metadata returned by the protocol. Multiple changes to the input data may be performed within the protocol.

- The term for referring to the identity is instantiated through the protocol and used as name for the identity by both parties. For ensuring uniqueness,

---

[18] Those tokens are obtained by the component by it accessing the data repositories and obtaining metadata on the opaque identity.

the term for the identity can be derived from the cryptographic object underlying the identity, for example, from the signature of an Identity Mixer private certificate. This approach is feasible as the term is only available in the output formulae.

- At $\mathcal{A}$'s side, the subject terms in the formula are replaced with the subject terms the party uses to refer to the subjects. This is important in order to ensure proper functioning of syntactical deductions using the data model including computing fulfillment of data requests. As $\mathcal{B}$ has initially created the input formula using its terms, there is no need to update the subject terms in $\mathcal{B}$'s formula.
- All predicates in $\phi'_{\mathcal{A}}$ referring to attributes of opaque identities refer in $\mathcal{A}$'s output formula to the values of those opaque attributes. Only those plaintext attributes are relevant when using the identity relationship, information on how they were obtained, that is, through this specific opaque identity, is not relevant for the subject in order to make use of the identity relationship. The attribute values are required to be known in the identity relationship of $\mathcal{A}$ in order to match it against policies for its use. Predicates related to opaque attributes can of course not be changed in $\phi'_{\mathcal{B}}$ as the attributes are hidden towards party $\mathcal{B}$.
- On-formula metadata predicates can be added or updated, e.g., for storing cryptographic values related to the identity relationship or identifiers thereof.
- The output formulae have new identifiers which is required as we operate in standard logic and objects cannot change through their lifetime.

The updated data on the side of $\mathcal{A}$ reflect the data to be used for expressing its identity relationship. The input data are not required any more after the protocol execution, yet can be retained for reasons of accountability. The above-outlined changes are protocol-dependent and thus performed by either the protocol engine or adaptor or a combination of both.

Example (Establishing an identity relationship (Output)): We continue our example with the output of an assumed successful protocol execution. Next, the output formula of $\mathcal{A}$ is given:

$$\phi'_{\mathcal{A}} = Eq(\text{id4890}.subject, \text{user4}) \wedge Eq(\text{id4890}.firstname, \text{Jane}) \wedge \ldots \wedge$$
$$Eq(\text{id4890}.ssn, 6789987667899876) \wedge$$
$$Eq(\text{id4890}.validfrom, 2009\text{-}07\text{-}01) \wedge Eq(\text{id4890}.validuntil, 2009\text{-}12\text{-}31)$$
$$Eq(\text{id4890}.certifier, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.subject, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.uniqueid, \text{german\_Government\_eID\_Issuer}) \wedge$$
$$Eq(\text{id4890}.protocolsuite, \text{Identity\_Mixer\_2048\_bit})$$

$\mathcal{B}$'s output formula is formed as follows:

$$\phi'_{\mathcal{B}} = Eq(\mathsf{id4890}.subject, \mathsf{user4567}) \wedge Eq(\mathsf{id4890}.firstname, \mathsf{Jane}) \wedge \ldots \wedge$$
$$Eq(\mathsf{id4890}.ssn, \mathsf{oid5678}.ssn) \wedge$$
$$Eq(\mathsf{id4890}.validfrom, \mathsf{2009\text{-}07\text{-}01}) \wedge Eq(\mathsf{id4890}.validuntil, \mathsf{2009\text{-}12\text{-}31})$$
$$Eq(\mathsf{id4890}.certifier, \mathsf{germanGovernment}) \wedge$$
$$Eq(\mathsf{cid7560}.subject, \mathsf{germanGovernment}) \wedge$$
$$Eq(\mathsf{cid7560}.uniqueid, \mathsf{german\_Government\_eID\_Issuer}) \wedge$$
$$Eq(\mathsf{id4890}.protocolsuite, \mathsf{Identity\_Mixer\_2048\_bit}) \qquad \square$$

Note that the terms used to refer to the identities are toy examples and would comprise more characters in a practical settings in order to realize the properties of uniqueness in the system with overwhelming probability. Also note the difference in the above output formulae: $\mathcal{A}$'s formula $\phi'_{\mathcal{A}}$ comprises the value 6789987667899876 for the social security number attribute while the output formula of $\mathcal{B}$ contains the relation between it and the *ssn*-attribute of the opaque identity as in the input. The other differences are the terms used for referring to the subject of the identity id4890: For $\mathcal{A}$, it is the term it always uses for referring to itself, for $\mathcal{B}$ it is the term already used in its input.

Once the protocol for establishing an identity relationship has been successfully executed, both parties store the data related to the identity relationship for further use: $\mathcal{A}$ needs to use it when releasing certified data based on this identity relationship to a data recipient; the certifier needs to use it when being involved in providing certified identity data based on the identity relationship to a data recipient and for a potential revocation of the relationship. That is, the identity relationship has a completely different meaning to the certifier and to the certifiee.

We want to note that the protocol can be equally used by $\mathcal{A}$ to establish an identity relationship for another party $\mathcal{S}$ with $\mathcal{B}$, e.g., in the case of $\mathcal{A}$ receiving the right to use attributes of the party $\mathcal{S}$ on the latter party's behalf in a delegation. For this, $\mathcal{A}$ will need to have an identifier relationship with $\mathcal{B}$ and establish one with subject $\mathcal{S}$ with $\mathcal{B}$. Only then, $\mathcal{A}$ can obtain an identity relationship issued for the subject referred to by the subject term for $\mathcal{S}$, and not $\mathcal{A}$ itself. This allows that a party $\mathcal{S}$ delegates the use of identifiers, attributes, or complete identities to a party $\mathcal{A}$ who can then use them on behalf of $\mathcal{B}$. When using them, $\mathcal{A}$ can make clear that it and $\mathcal{S}$ it is talking about are different subjects by using identifiers appropriately.

9.6.4.4 ReleaseData

The protocol *ReleaseData* is used by a party to release data based on one or more identifier relationships and identity relationships to another party. An instance of this protocol is triggered by the party who intends to release data, e.g., a user interacting with a service provider. Within the overall architecture, the *ReleaseData* protocol is mainly invoked from within

the policy-driven negotiation protocol of Section 9.9 for mutual release of certified data.

The protocol for releasing data is a protocol between a party $\mathcal{A}$, the data provider, and a party $\mathcal{R}$, the data recipient. At the time of starting the protocol, $\mathcal{A}$ knows $\mathcal{R}$ at least by a pseudonymous identifier $s_{\mathcal{R}}$. Note that the latter may default to $\mathcal{R}$'s public identifier (pseudonym) in case of $\mathcal{R}$ being a service provider with publicly-known identifier. Mainly in the case of $\mathcal{R}$ being a user, allowing it to establish a pseudonymous identifier $s_{\mathcal{R}}$ with $\mathcal{A}$ is useful for two-way pseudonymous interactions between users.

Much like the other protocols discussed above, the input of this protocol is also based on our data model to a large extent. Next we present the API of the protocol in detail.

ReleaseData

Input	
$\mathcal{A}$	$\phi_{\mathcal{A}}, s_{\mathcal{R}}, \{(o_i, \{\nu_{l_{\mathcal{A},i}}\}_{1 \leq l_{\mathcal{A},i} \leq n_{\mathcal{A},i}})\}_{1 \leq i \leq r_{\mathcal{A}}}$
$\mathcal{R}$	$-$
Output	
$\mathcal{A}$	$\phi'_{\mathcal{A}}, \{(o'_j, \{\nu_{l'_{\mathcal{A},j}}\}_{1 \leq l'_{\mathcal{A},j} \leq n'_{\mathcal{A},j}})\}_{1 \leq j \leq r'_{\mathcal{A}}}, success_{\mathcal{A}}$
$\mathcal{R}$	$\phi'_{\mathcal{R}}, \{(o'_k, \{\nu_{l'_{\mathcal{R},k}}\}_{1 \leq l'_{\mathcal{R},k} \leq n'_{\mathcal{R},k}})\}_{1 \leq k \leq r'_{\mathcal{R}}}, success_{\mathcal{R}}$

The input comprises a data statement $\phi_{\mathcal{A}}$, an identifier $\mathcal{R}$ is known to $\mathcal{A}$ under, and a set of metadata predicates for each identifier object and identity referred to from within $\phi_{\mathcal{A}}$ as an input of party $\mathcal{A}$. Party $\mathcal{R}$ does not provide input in this protocol as $\mathcal{A}$ completely determines the data to be released to $\mathcal{R}$. This is the natural API of a protocol for releasing data. The output of $\mathcal{A}$ comprises a new data statement $\phi'_{\mathcal{A}}$ based on $\phi_{\mathcal{A}}$ as well as a set of (updated) metadata predicates returned as output at the side of $\mathcal{A}$; $\mathcal{R}$'s output comprises, analogously, a data statement $\phi'_{\mathcal{R}}$ based on $\phi_{\mathcal{A}}$ and metadata predicates returned as output on the side of party $\mathcal{R}$. The output of both parties comprises a boolean flag *success* indicating the success of the protocol.

The data statements ϕ express, similar as in the protocol for establishing an identity relationship, attribute data and in-formula metadata such as temporal validity and certifier metadata. See the protocol for establishing an identity relationship and Section 9.3 on the data model for details. Clearly, the meaning of the formula in the *ReleaseData* protocol is different to the one of the other protocols. The expressivity in terms of permitted predicates is usually different between the protocols for establishing an identity relationship and releasing data due to the technology being used, e.g., private certificate systems.

Example (Data formulae of a release protocol): The following example formula is a formula created by the releasing party $\mathcal{A}$ to be released to party $\mathcal{R}$. We use the identifiers, identities, and terms from previous examples. Note that the terms used for $\mathcal{A}$'s identities as well as the subject term to refer to $\mathcal{A}$ are the ones as used in the identifier and identity relationship formulae of $\mathcal{A}$ for the used identifiers and identities, that is, have not (yet) been renamed.

$$\phi_{4765} = Eq(\text{id4890}.subject, \text{user4}) \wedge Eq(\text{id4890}.firstname, \text{Jane}) \wedge \ldots \wedge$$
$$Leq(\text{id4890}.dateofbirth, \text{1992-02-01}) \wedge$$
$$Leq(\text{id4890}.validfrom, \text{2010-02-01}) \wedge Geq(id7560.validuntil, \text{2010-02-01}) \wedge$$
$$Eq(\text{id4890}.protocolsuite, \text{Identity\_Mixer\_2048\_bit}) \wedge$$
$$Eq(\text{id4890}.certifier, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.subject, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.uniqueid, \text{German\_Government})$$

The example specifies that the first name attribute of the identity id4890 of party $\mathcal{A}$ is released, as well as a proof that the date of birth dates back at least 18 years from the time of creation of the formula. The in-formula metadata attributes *validfrom* and *validuntil* are used to establish the temporal validity of the identity id4890. The certifier is uniquely specified using the usual (fixed) term to refer to it. The protocol suite is specified to be Identity_Mixer_2048_bit, the same one used in the protocol for establishing the identity relationship. The name ϕ_{4765} of the formula is chosen randomly or created through a one-way function on the formula.

The following is a new formula ϕ_{6547} with terms having been renamed such that the formula can be sent to the intended recipient party without any compromise of privacy due to unintentional linkability through term equality. This new formula is generated by the protocol engine of $\mathcal{A}$ through renaming of certain terms according to rules dependent on the protocols underlying the identifier objects and identities used to build the formula. The example assumes the Identity Mixer protocol suite being used for release of data based on a private certificate.

$$\phi_{6547} = Eq(\text{id5160}.subject, \text{user3982}) \wedge Eq(\text{id5160}.firstname, \text{Jane}) \wedge \ldots \wedge$$
$$Leq(\text{id5160}.dateofbirth, \text{1992-02-01}) \wedge$$
$$Leq(\text{id5160}.validfrom, \text{2010-02-01}) \wedge Geq(\text{id5160}.validuntil, \text{2010-02-01})$$
$$Eq(\text{id5160}.protocolsuite, \text{Identity\_Mixer\_2048\_bit}) \wedge$$
$$Eq(\text{id5160}.certifier, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.subject, \text{germanGovernment}) \wedge$$
$$Eq(\text{cid7560}.uniqueid, \text{German\_Government})$$

A noteworthy point to mention is that the new term referring to the identity is id5160. The term referring to the subject attribute value has been renamed to user3982 and must be either the same term used as subject of an identifier object established with the intended recipient of the formula or a freshly-chosen term in case the data release is not to be linked with an identifier between the parties. The term used to refer to the certifier's identity is cid7560 as in the input formula ϕ_{4765}, assuming the standard case of the certifier using a publicly-known identity towards other parties. The formula just above is the protocol output of both parties $\mathcal{A}$ and $\mathcal{R}$: It is the actual data statement sent from $\mathcal{A}$ to $\mathcal{R}$, accompanied with a proof of its correctness. □

When using the Identity Mixer protocol for proving the integrity of the formula with respect to the referred to identifier and identity relationships, the

new terms for referring to objects and the subject are determined by the cryptographic protocol and cannot be arbitrarily chosen by the party. This leads to terms that are unique and fresh with overwhelming probability and lead to a collision with already-existing terms with negligible probability. From a modeling perspective this is an ideal situation as the terms are provably correctly chosen which may not be the case when different or no cryptographic schemes are used. Concretely, party $\mathcal{A}$ generates a term for an identity through a cryptographic hash of the cryptographic values used in the protocol corresponding to the identity to be named. For a subject identifier, the part of the cryptographic proof related to the cryptographic pseudonym and the party's master secret value are used to generate the identifier. Due to the properties of the Identity Mixer protocol suite, the uniqueness property holds with overwhelming probability for the derived identifiers.

A data statement can make use of multiple identity relationships and thus require multiple different underlying protocols for the actual exchange of the data, e.g., a credential system and a plaintext release protocol of attribute data. This is allowed by our architecture and handled by the orchestration component: The component may need to split the data formula into multiple formulae and execute protocols for the release accordingly, e.g., a credential protocol and a plaintext release protocol to release a compound statement comprising both identity information certified by third parties and claimed by the party. A data statement can also build on multiple formulae of different identity relationships with mutually "compatible" protocols. In this case, identities of different identity relationships can be related to each other, e.g., their subjects can be expressed to be the same party without revealing its value. Currently, private certificate systems are the only technology supporting such proofs that involve multiple identity relationships, even with different certifiers, and even allow to relate unrevealed attributes of such to each other. For claims this is trivially achieved, though based on honesty of the claiming party and thus not interesting.

Once a protocol for releasing data has been successfully executed, the releasing party stores the released formula as well as any associated metadata in its data track related to itself as subject while the recipient stores the received formula and metadata in a profile data entry with the subject being the term the sender used in referring to the releasing party. The recipient's metadata, depending on the protocol used, comprise (cryptographic) proofs showing the correctness of the released formula as well as evidence objects related to the conditionally-released identities. The latter allow a third party to obtain the attribute values of the conditionally-released identity if the attached condition is fulfilled. This post-processing is performed by an orchestration component that has triggered the protocol for releasing data.

When relating multiple identifier objects or identities with the subject attribute and using the Identity Mixer private certificate system for proving the given formula correct, using the same subject for two different objects means that the subject party of those is the same, endorsed by the same master

secret for the subject being used for the objects when generating the proof. This is strong evidence of sameness as a party can be disincentivized or prevented from sharing its master secret key through different technical or organizational means or a combination thereof. See Bichsel [Bic07] for a discussion of sharing prevention and references to relevant literature. An exception to this are identity relationships obtained in a delegation of identity as in this case the subject of an identity can be another party, not the party making the statement about it.

9.6.4.5   RevokeIdentityRelationship and FinalizeConditionalRelease

We do not give the details for the remaining protocols *RevokeIdentityRelationship* and *FinalizeConditionalRelease*, but only present the overall ideas of those. The basic idea behind the protocols is that they allow that the attribute values of a conditionally-released identity received and held by a data recipient $\mathcal{R}$ can be obtained through involving a conditional data recipient $\mathcal{T}$ the conditionally-released identity has been targeted at during release. The trustee may obtain the attribute values once this is requested and the condition attached to the identity is fulfilled.

9.6.5   Components

We now give more details on the architecture for the data exchange component, see Figure 9.4 for an illustration of its internal architecture.

9.6.5.1   Data Exchange Orchestrator

The purpose of the data exchange orchestrator component is to dispatch a request arriving at the data exchange component to the adaptor capable of handling such a request. This component particularly allows for different protocols being used for proving parts of a single identity formula provided as input. In this role, the component pre-processes and dispatches calls to the data exchange component towards the appropriate protocol adaptors. Particularly, it splits the input into its different parts that can each be handled by a single invocation of a protocol adaptor. Vice versa, when the calls to the adaptors return, it assembles the returned parameter of the data release component from various sub-responses provided by the protocol adaptors that have been invoked. The component operates on the data representation that is used at the interface of the data exchange component.

9.6.5.2 Protocol Adaptor

We next investigate the role of a protocol adaptor in some more detail. As mentioned above, an important design characteristic of our architecture is that at the level of the data exchange component the interface is generic, that is, independent of the used protocol. For each supported protocol, the protocol's specific interface is exposed by the corresponding protocol engine. The glue between the generic interface of data exchange in general and the specific interface of each data exchange protocol suite is done through a protocol-specific *protocol adaptor*. This adaptor has the required "knowledge" and functionality of performing a bi-directional mapping between the generic interface of the data exchange component and the specific interface of the protocol engine it is associated with. Thus, an instantiation of this component needs to be implemented for each new protocol to be integrated.

The concrete tasks of a protocol adaptor depend on the protocol engine it is associated with and may vary widely depending on functionality and type of the protocol.

- **Data translation:** It maps between our data model and the data representation used by its protocol if the protocol deviates from the data model. This depends on the implementation of the protocol engine and a mapping being required will be the prominent case; in many envisioned protocol engines at least some differences in data representation are implied by the fact that aspects such as key management are handled by the adaptor. It is important to note at this point that our data model facilitates this approach by the semantics of parts of it being used throughout the architecture, e.g., in the matching of a policy against a party's identity relationships regardless of the protocols underlying the relationships, and other semantics being used by specific elements in the architecture, e.g., the Identity Mixer library and its adaptor. The mapping can, in the trivial case, be the identity function. An important part of the data translation is the mapping between the data model and key and token identifiers.
- **Term mapping:** The mapping of terms from the party's terms used for objects such as identifier objects, identities, or subjects to the terms used when talking about those towards other parties is performed by this component. The mapping is dependent on the protocol being used, e.g., the term names may be determined at the cryptographic level as in the case of Identity Mixer or may be randomly chosen. The mappings are stored through metadata associated with data being handled.
- **Token management:** Cryptographic tokens such as cryptographic pseudonyms, cryptographic commitments, or private certificates can be maintained (stored, retrieved, updated, and deleted) by the adaptor. The architecture is flexible on how this is done, e.g., by accessing an encrypted storage component or secure hardware tokens. Alternatively, the cryptographic library can itself perform the storage of such tokens or the tokens can be passed back as binary metadata inside of metadata predicates that

are not interpreted by other parts of the architecture and transparently stored with the identity relationship.

- **Key management:** For protocol engines that require cryptographic keys, which is true for almost all protocols of interest in a privacy-enhancing IdM system, the engine may rely on the outside world, that is, the adaptor or another component in our case, to manage the keys. Key management is a special part of token management in that keys need to be stored and loaded once they have been obtained and verified. Key management includes retrieving locally, or remotely and verifying public keys required for processing a formula at hand. In case the engine manages its keys internally, the adaptor must map the key information extracted from the data statement to key identifiers understood by the protocol engine.

- **Update of state:** Certain state information may need to be updated, triggered by external events. An example for this in the area of the Identity Mixer system is the update of revocation information stored with a private certificate: For a revocation scheme for private certificates [CL02], a certificate needs to be updated whenever any certificate issued by the same certifier key has been revoked in order to be able to still use the certificate. Such an update can be either triggered asynchronously by the protocol adaptor, independent of any interaction being performed, or once a certificate is being used. For better privacy, such a process should not be triggered by a related action, e.g., the use of the concerned private certificate, but independently to not establish undesired linkability.

For both key and token management, the architecture is, as it does not impose a specific approach, sufficiently flexible to allow one to leverage existing solutions for key management and token storage and integrate them into the architecture or alternatively build one's own cross-protocol or protocol-specific solution.

Depending on the protocol engine, there may be further tasks needed to be performed by the adaptor in order to integrate a protocol engine into our architecture. As we cannot anticipate all protocol engines, it is infeasible to list all those tasks, though, we think to have captured the most prominent ones in our discussion above.

9.6.5.3   Protocol Engine

A protocol engine provides the implementation of the protocols required for a concrete protocol suite (i.e., scheme or algorithm) for data exchange. There are no specific requirements on the interface of the protocol engine – a protocol adaptor is used to translate between it and the uniform interface of the data exchange component. A protocol engine can even be a heavyweight system itself that is accessed through a Web Services interface and resides on a separate machine as may be the case for commercial products for the server side.

The protocol engine may carry out the communication with other parties itself to run protocols, or they may expose a state-machine-based interface that is driven by the protocol adaptor, thus communication being performed by the protocol adaptor, e.g., via a dedicated or shared communication component. We do not restrict the architecture in this respect in order to not exclude protocol engines from being integrated into our data exchange architecture.

For the Identity Mixer protocol, we have taken the design decision that the protocol engine is a state machine that is "driven" by an outside component to consume and generate protocol messages until the protocol engine indicates termination of the protocol. The outside component is then responsible for performing the communication with the other party. This design decision is motivated by a separation of the handling of the channel and the implementation of the cryptographic protocol which we decided to keep separate. This approach may avoid extra round trips on the network and can integrate the communication related to data release protocols and other protocols into a single communication channel.

9.6.6   Aspects of System Architecture

This section explains additional aspects relevant for the data exchange component as well as aspects on how to connect the component to the overall architecture. We particularly focus on specifics of private certificate systems.

9.6.6.1   Certificate Structure

Regarding the definition and use of private certificate systems, there is a substantial difference to other technologies for data exchange. In contrast to certificates based on traditional signature schemes like RSA or DSA, a private certificate does not necessarily encode certain information elements needed to link the certificate's attributes to our data semantics in an integrity-protected way. Thus, for a private certificate system an additional information entity is needed: A *certificate structure* defines the internal structure of a private certificate as well as its features, and a mapping between the abstract high-level data representation of our data model and the concrete technical representation, namely signed tuples of integers. Concretely, the certificate structure includes the ontology types and data types for the attributes, a mapping of attributes expressed in the data model to one or more integer attributes, and a mapping of certificate features to integer attributes. This low-level technical information is required in addition to what is specified through our data model within the identity relationship the private certificate is associated with in order to execute protocols based on a private certificate. This aspect of the implementation of a private certificate system has first been discussed by Camenisch et al [CSZ06].

In traditional attribute exchange technologies, e.g., X.509 certificates, such information is already included in the certificate itself and signed together with

the attribute data using traditional signature schemes such as RSA or DSA, i.e., all together is considered a single message for the signature scheme. As private certificate systems use so called block signature schemes such as the SRSA-CL [CL03] or BL-CL [CL04] schemes which are used to sign tuples of integer attributes without such metadata – the metadata must be maintained in addition to the message tuple of integers, and each attribute, regardless of its type, must be encoded in one or more integers of the certificate. We use the certificate structure for storing this metadata for a private certificate.

A certificate structure must be issued by the certifier for each class of private certificates it issues, where the class is defined by comprising the same attributes of the same type. This immediately implies the independently expressed requirement that certificates of a certain type must all conform to the same certificate structure in order to be valid certificates of this type.

For reasons of data integrity and thus security, a certificate structure must be integrity protected, that is, integrity must hold towards all parties who use certificate structures in their private certificate protocols. In case of integrity being compromised, a rogue prover could exchange ontology and data types as well as encoding order for its certificates and, with a well-crafted attacker's certificate structure, obtain authorization for access to resources it is normally not authorized for.

Integrity can be achieved via any of multiple possible means – we show the two most practical ways of how to protect integrity of certificate strucutures in our architecture. As one way, the certifier can sign the certificate structure for a certificate with its signing key pair. This signature can be easily obtained and verified using standard means by any system participant. Alternatively, the certificate structure can be included as a special attribute into each private certificate of this type the certifier issues. This is done through representing the certificate structure as an integer value by applying a cryptographic hash function such as SHA-256 on it and encoding the result as an integer attribute in the certificate. The information regarding the certificate type can then be distributed without further consideration of integrity. This approach allows the subject of the certificate as well as each data recipient the subject releases data to by using the certificate to validate the integrity of the certificate structure and thus securely verify the proof of the subject.

The first approach presented of maintaining integrity works well with certifiers that are identified, though, is not applicable to anonymous certifiers. In such settings, the alternative approach is preferable. Our architecture supports both approaches to remain open for future applications of anonymous certifiers.

9.6.6.2   Templates for Identity Relationships

An important architectural problem that deserves discussion in a comprehensive treatment of a data exchange architecture is how a certifier obtains the

data to vouch for about a subject, and particularly how the process of establishing identity relationships is integrated with the overall system architecture. In short, this addresses the question of how the input to the data release component is obtained for the protocol of establishing an identity relationship. We note that for the data exchange component it is assumed that this information is received as input from other components in the architecture. We next discuss a practical and flexible solution for how to generate this input. Our approach allows a certifier to include arbitrary external data sources to obtain the data from in addition to data provided by the other party in the interaction or previous interactions.

We start the discussion with the observation that the data a certifier vouches for is held by the certifier or obtained by the certifier prior to allowing an instance of the *EstablishIdentityRelationship* protocol to be triggered. One possibility for starting the protocol is that the process of establishing an identity relationship is triggered by an action of the requester, such as clicking a Web link for obtaining the course completion certificate of a uniquely-identified on-line course for which she has successfully taken the exam. After such a triggering event, the certifier (e.g., an e-learning provider) needs to start the *EstablishIdentityRelationship* protocol with the appropriate attribute data and parameters for the new identity relationship. We address exactly the issue of how those data are obtained and the input to the data exchange component is created.

We build on the concept of *templates* for identity relationships as a simple and general way of realizing the discussed functionality. A template is a data representation, including an "uninstantiated" formula in our data model as well as associated on-formula metadata. The formula represents the class (set) of all formulae of identity relationships of the same type that can be issued by the certifier. The template itself does not represent instance data, but rather is instantiated with concrete data by the certifier for each new instance of the *EstablishIdentityRelationship* protocol the template is used for. One template is maintained by a certifier for each *type* of identity relationship it vouches for, where the type is the one of the core identity of the relationship.

The data formula $\phi_{\mathcal{T}}$ of a template expresses identities through variables, the attributes of the identities are encoded as constants as usual, but each attribute value of the main identity is encoded with a special symbol θ with an annotation. The symbol indicates that no value is present for this attribute in the formula, but rather specifies through its annotation how the value can be obtained. The vocabulary of the annotations is deployment specific and determines for the party how to obtain the data. For example, the annotation can indicate that the attribute of the new identity relationship it represents should be pulled from a local LDAP directory server or an SQL data base. Further information for obtaining the attributes may come from the dynamic subject context for the requester. The context may, e.g., contain the pseudonym or other attributes to use for addressing the data source.

The on-formula metadata of the template comprises the usual metadata required for establishing an identity relationship which may depend on the underlying protocol being used. In case of the Identity Mixer protocol, it comprises the certificate structure as explained in the subsection above as an important item.

9.6.6.3   Architectural Integration of the Protocols

We next discuss how the different protocols are integrated into the architecture in terms of how, in which contexts, and by which party they are triggered. This should convey to the reader the bigger picture of the data release component within the architecture.

Creating an Identifier

The protocol for establishing an identifier relationship is triggered by party $\mathcal{A}$, party $\mathcal{B}$ is listening for protocols of this kind and triggers one when a message from the communicating party comes in. The decision to trigger such a protocol can either come from party $\mathcal{A}$ or party $\mathcal{B}$. The further case is when party $\mathcal{A}$ decides to establish a new pseudonym, e.g., to start a new message exchange with $\mathcal{B}$ – possibly unlinkable with previous message exchanges with this party. The latter is the case when $\mathcal{B}$ decides, e.g., through its business logic, that it needs a (new) identifier relationship by the other party. This is typically the case when $\mathcal{B}$'s business logic decides that it needs a pseudonym of $\mathcal{A}$, e.g., for the creation of a pseudonymous account. In any case, $\mathcal{A}$ can be given the possibility of consenting to the establishment of the identifier relationship, though, for usability reasons this can be hidden from a user as running such a protocol does not include a release of identity data about $\mathcal{A}$.

Creating an Identity Relationship

The protocol is triggered by the certifier $\mathcal{B}$ as it is the party that decides on the input to the protocol, i.e., chooses the attribute data to be vouched for and decides that it intends to initiate such a relationship. As in the case of the protocol for establishing an identifier relationship there are two ways on how the protocol is triggered practically. The certifier can trigger the protocol from its side and thus initiate it without preceding communication on this with $\mathcal{A}$, or $\mathcal{A}$ can request the issuing, e.g., by following a corresponding Web link which then leads to $\mathcal{B}$ starting the protocol. In either case, the protocol endpoints execute the exact same protocol flow. Note that optionally $\mathcal{B}$ may provide the data formula specifying the to-be-established identity relationship before invoking the protocol. Then $\mathcal{A}$ can give consent based on this specification. As another option, $\mathcal{B}$ can, alternatively, provide details on the identity relationship to be established only in an informal way, e.g., on the Web page presented to the other party together with a Web link for obtaining the identity relationship.

In either of the above cases, the establishment of an identity relationship is seamlessly integrated into the architecture by modeling it as a resource access by the party requesting the establishment of the new identity relationship. A resource identifier is used for representing the protocol and one or more authorization policies are defined on the resource as can be done for any resource in the system. The authorization policies specify all data items that are needed in order to create the new identity relationship, in addition to data that may be held already locally by the party. Particularly it requires an identifier the requester is known under at the party (pseudonym), and may require attribute information, and opaque identities the parties have previously exchanged.

The resource is associated with a template $d_{\mathcal{T}}$ for an identity relationship comprising the formula $\phi_{\mathcal{T}}$ and metadata. The formula $\phi_{\mathcal{T}}$ can refer to identifier objects, identities and opaque identities using the terms for variables as used in the authorization policies that are defined on the resource. Using the same term means that the object referred to by this term in the policy and the template is the same one following standard interpretation rules of first-order logic. This approach allows the certifier to relate the new identity relationship and the data provided by the party for fulfilling the policy in a simple yet effective manner, fully driven by the policies of the party. For any other attributes to be used in the creation of the identity relationship, the template refers to arbitrary data sources accessible by the party. The party needs to implement the mechanisms to pull the data from the data sources which may be proprietary or based on standards. In summary, the formula $\phi_{\mathcal{T}}$ of the template is a semi-formal construct that becomes a formal data representation formula ϕ of our data model once instantiated with concrete attribute data obtained as specified. The components used to implement the (connectors with) components for providing attribute data are deployment specific and not further discussed here.

Releasing Data

The protocol for releasing data is triggered by party $\mathcal{A}$ because it is the party that decides on the data to release and to release data to the other party. Thus, our design decision, that the protocol is initiated by $\mathcal{A}$ reflects the nature of the protocol. The protocol can, much like the other protocols, be executed as sub-protocol of other protocols. The common case will be the one of the surrounding protocol being an instance of the negotiation protocol of Section 9.9. In this case, multiple instances of the *ReleaseData* protocol can be executed sequentially within a single instance of the protocol for releasing data.

9.7   Authorization Policies

Defining authorizations for access to resources of a party by other parties is a prerequisite for an open system in a real-world setting in which not all parties are fully trusted. An *authorization policy* specifies authorizations for the access to resources in a formal language. For our architecture, the resources are data and services, but we do not restrict other resources from being referred to. Our architecture is targeted at interactions between parties like users and service providers, with the goal of giving the one party (e.g., the user) access to services of the other party (e.g., the service provider) while authenticating the requesting party via (certified) attribute information. The goal is to allow the requester to access services or data while remaining anonymous or pseudonymous instead of requiring it to be identified as usually done in today's systems. We anticipate that authorization policies form the core policy system of our approach to negotiation, a mutual exchange of data and agreement on data handling policies between two parties – details on this are presented in Section 9.9.

9.7.1   Paradigms of Authorization Systems

Over time, various paradigms have evolved for authorization systems. The traditional paradigms build on *access control lists* or *access control matrices*, specifying which subjects of the set $S = \{s_1, \ldots, s_{k_S}\}$ of subjects have access to which resources of the set $R = \{r_1, \ldots, r_{k_R}\}$. Both subjects and resources are specified through their (unique) identifiers s_i and r_j, respectively. An access control list (ACL) for a resource r_j is a list specifying the subjects $\langle s_{i_1}, \ldots s_{i_n} \rangle$ that may access the resource r_j. An access control matrix is a matrix with the subjects and resources making up a dimension of the matrix each, and the cells s_i, r_j of the matrix being the permissions of subject s_i on resource r_j. Authorization is done based on the authenticated identifier of the subject. Clearly, the authorization engine (or the responsible party) can create a complete profile of who has been accessing which resources.

Such traditional approaches are still extremely well suited for closed systems managed by a single domain of control, such as the file system of a multi-user computer, but are immediately ruled out for an open distributed system such as the Internet where users from different domains of control need to be authorized for performing operations on resources held by parties in different domains of control and where privacy protection is a requirement.

A main problem with the traditional approaches is the identifier-based approach that does authorization based on identifiers of parties. In open systems, authorizations are often not granted based on the identifier of the subject, but rather its properties. As an example, consider any service subscription scenario in the Internet: In such a scenario, the relevant property of authorizing an access request is the holdership of a service subscription by the subject, regardless of the identifier or other identity attributes of the subject.

The paradigm of *attribute-based authorization* makes authorization more open because subjects can be specified through their attributes and not identifiers. Attributes can be properties of the person such as parts of their civil identity as well as permissions assigned to them. In this paradigm it is not a requirement any more that a subject identifier be authenticated and used for the authorization decision and thus the subject be identified among all subjects in the system. In fact, the subject can be anonymous in the set of all subjects of the system having the same attribute values as revealed in a transaction, the so-called anonymity set.

9.7.2   Our Approach

We build on top of the attribute-based paradigm of authorization and take the ideas further than previous approaches in the literature. Thereby we build on top of the access control system of Bonatti and Samarati [BS02b] and its extension by Ardagna et al. [ACDS08]. Their system is a "privacy-aware access control system" for multiple of its properties: a policy allows for specifying the subject (requester) through its attributes without requiring it to be identified; the language supports purpose binding of an access request; the system has been designed with user control of the user's identity in mind; the extended version of the system in [ACDS08] integrates aspects of data handling for secondary use of data directly into the language; and the system is open in terms of its architecture and policy language, thus is targeted at large distributed systems. In the remainder of this section, we present aspects of this system and particularly the language, and of our extensions thereof, which are relevant for our discussion in the contect of our architecture. See Chapter 11 of this book for a detailed description of the authorization system we build on.

The setting we operate in is the usual setting already shown in Figure 9.2. When focusing on the authorization-related aspects, the setting can be simplified by reducing it to the user (or subject or requester) and the service provider as the active entities to be considered. Both subjects and service providers have authorization policies defined on their resources. A service provider's resources are services as well as data pertaining to themselves and data pertaining to other parties, such as its customers. A subject has authorization policies defined on her data and possibly data she holds about other parties.

For our extensions of the language and model as well as the integration into the architecture, we have taken the approach of pushing privacy further in terms of expressivity of the language and policy model. Specifically, our authorization policy language has been defined with private certificate systems in mind as the underlying data exchange mechanism. We leverage formulae expressed in our data model as language elements and thereby immediately obtain the advanced expressiveness in terms of privacy. Doing so allows for leveraging many of the privacy features of private certificate systems and deliver them to the parties in a system based on our architecture.

We deliberately took the approach of not integrating the policy model fully with our data model for which a formal semantics exists and thereby also providing an integrated formal semantics for the policy model, the reason for this being that this allows the architecture to be applicable to different policy models without a change of its formal foundation of data representation as well as a simpler data semantics.

9.7.3 Language Basics

We give an overview of the language for expressing authorization policy rules, being an extension of the language of Bonatti and Samarati [BS02b] and Ardagna et al. [ACDS08]. As in the original work, an authorization rule is an expression of the following form:

> *subject* [WITH *subject_expression*] CAN *actions*
> ON *object* [WITH *object_expression*]
> FOR *purposes* [IF *conditions*]

Note that subsentences contained within brackets [...] are optional elements of the policy rule. The elements of a rule are defined as follows, based on [ACDS08]:

1. *subject* identifies the subjects (parties) to which the rule refers, that is, to whom the authorizations defined by the rule apply. It corresponds either to a set of identifiers of parties or a concept in an ontology, such as a class in an abstraction hierarchy. The special keyword *any* is used to denote that the rule refers to any party.

2. *subject_expression* restricts the set of subjects that the rule applies to. It is a formula in our data model. The formula specifies a set of subjects through the data statement expressed in the formula: Each party who can fulfill the formula (with its attribute data) is contained in the set of parties the subject expression defines. Note that in practice, when specifying attribute-based access control rules, the subject is often set to *any*, thereby not imposing restrictions, and the set of parties the rule applies to is completely specified through the *subject_expression* element. If both the *subject* and *subject_expression* are defined, the subject expression refers to a subset of the set defined by the subject.

3. *actions* is the set of actions to which the rule refers. Examples of actions are, without a claim for completeness, *read*, *create*, *write*, and *delete*. This set of actions can be changed for concrete systems.

4. *object* identifies the objects to which the rule refers. It corresponds either to a set of identifiers of objects or a concept in an ontology, such as a class in an abstraction hierarchy. The keyword *any* denotes that the rule refers to any object. This is analogous to the *subject*.

5. *object_expression* restricts the set of objects the rule applies to. It is a formula based on our data model. The formula refers to a set of objects through the statement expressed in the formula: Each object whose object profile fulfills the formula is contained in the set specified through this expression.
6. *purposes* defines the purposes related to the request. It is a set of purposes with the concrete purposes to be defined for a system.
7. *conditions* is a boolean formula specifying conditions that must hold. Such conditions can be expressed on generic predicates that are evaluated at the time of access. The formula may comprise elements of our data model as well to express conditions over identity and object data.

Our main contributions in the area of authorization languages are twofold: A change of the rule language and policy model based on the original work and an integration of the resulting policy language and model into our architecture. Both the extensions and the integration have been performed with the goal of increasing the expressive power of the language and overall resulting system for strengthening data privacy of the users. We note that the original scheme ([ACDS08]) fits into the architecture as one specific less powerful embodiment, though, requires additional mappings to our data model; our extended model represents a more powerful embodiment, in parts based on experience gained from the process of building the PRIME Architecture. See Chapters 11 and 14 of this book for details on the original policy system that has been implemented in the PRIME prototype built during the project and its implementation-level architecture.

9.7.4 Language Extensions

An important change of the language affects the subject expression. The subject expression specifies the requirements on the requester (subject) in terms of attribute data. The original language has multiple aspects that are worth to be improved for obtaining stronger privacy properties: The language expresses certified data through the use of so-called "credential terms" (predicates) without the possibility of stating that multiple attributes must be contained within the same credential, that is, be grouped together. Such a grouping of attributes is relevant in many circumstances, e.g., when modeling credit cards, the credit card number and expiration date must be associated to the same credit card, or when modeling a bank account, the account balance and currency must be grouped. As another weakness, the language specifies the certifier of data in a credential term only through its public key, that is, refers to a concrete uniquely-identified party, which constrains the expressiveness and therefore the suitability for large-scale open systems where one needs to express the certifier in more generic ways, e.g., through its attributes (properties), also taking ontologies into consideration. A further weakness is the lack of language support for achieving conditional accountability for anonymous transactions

as well as the general lack of language elements for realizing more aggressive privacy-enhancing features.

9.7.4.1   Subject and Object Expression

We improve the language by allowing a subject expression to be specified through a formula in our data model. The data model has been crafted for applicability to both representing data (locally at a party and in interactions) as well as expressing requirements of requesters in policies as shown in Section 9.3. Those aspects are closely related in their nature as a policy rule specifies (parts of) a request for data while a data statement is made by a party in response to such a request.

Through the use of our data model for representing subject expressions, attributes can be grouped into named sets of attributes (identities) thereby resolving the mentioned problem of illegitimately combining attributes of different identities.

As another major feature of our data model, the certifier can be specified through predicates over an identity it is the subject of which is a major extension towards the openness of the system. This allows one, for example, to express a set of possible certifiers through their abstract properties, instead of by an identifier (or key). The original language [ACDS08] specifies the certifier through an identifier which makes the language is less expressive than ours.

Through the feature of our data model of supporting conditionally-released identities, our language allows for accountability while retaining anonymity of transactions. This balance between privacy and the possibility to—under clearly defined circumstances—revoke the anonymity of the subject, is in our view a necessary feature of a language to be deployed in a practical environment. When considering current trends in legislation and society, it may well be that unconditional anonymity will be increasingly less appreciated for real-world systems in the mid-term future.

For further details and advantages of our extensions to the subject expression, we refer the reader to Section 9.3 on the data model. To summarize, our subject expression is a logic formula in a fragment of first-order logic and the concepts introduced in our data model carry over directly to the subject expression. As a final note, we want to stress that the subject expression of an authorization policy is formulated using the language parts for requesting data of our data model.

The object expression in the original language is a boolean formula with the same syntax as the subject expression. The object expression refers to a set of objects by making statements over their object profiles, that is, metadata associated with the objects. Particularly it allows for relating object attributes to subject attributes by using the keywords *subject* and *object* as placeholders for the subject and object profile, respectively. We express the object expression with a formula in our data model and thus immediately apply its advantages to the specification of object expressions, much like we do

for subject expressions. As presented next, we improve, as a specific feature, the expressivity of dependencies between subject data and object expressions to allow for stronger data minimization than in the original model.

9.7.4.2   Subject-Dependent Object Expressions

For realizing a policy that specifies its applicability to an object by conditioning the object profile on data about the subject can greatly simplify the specification of policies for important use cases. Take as an example a secret agency that needs to restrict access to confidential files according to security clearances: A subject needs to have at least the same clearance as the resource she wants to access, the clearances being expressed as integers. As an online shopping example, take an online movie store which needs to make sure that a customer can only download or stream a film when she satisfies the age requirements for the film. As an example from the healthcare environment take the one of a treating physician having full access to the patient data of her patients. An increasingly-discussed use case is pseudonymous on-line subject access to data, that is, a subject may access their own subject profile at the service provider online, but no one else's.

Those examples have in common the goal of specifying the mentioned restrictions in a single policy for all resources in question instead of a separate policy covering each resource. The latter would be trivially doable by simply specifying the requirement in the subject expression individually for each resource. Multiple drawbacks are related to this trivial approach: An inflation of the number of policies to be defined by a party, even though this can be done automatically, resulting performance degradation of the system, and resulting harder and less transparent policy management.

Example (Agency): Let a classified object r of interest to a subject have a clearance level of 3, meaning that any requester needs a clearance of at least 3 to access the object. This is expressed through the value assignment $Eq(\mathsf{object}.securityClearance, 3)$ in the object profile for r. Let furthermore the policy of which a fragment is shown be specified for all classified objects of the agency through means of abstraction or enumeration:

$$any \ \mathrm{WITH} \ldots \wedge Eq(\mathsf{C}.status, \mathsf{Special\_Agent}) \wedge \ldots$$
$$\mathrm{ON} \ldots \mathrm{WITH} \ Geq(\mathsf{C}.securityClearance, object.securityClearance)$$

This fragment of a rule expresses that only special agents having at least a clearance level as the object can access the object. At access time, the following fragment of a data request is compiled from the policy rule to be sent to the subject:

$$\ldots \wedge Eq(\mathsf{C}.status, \mathsf{Special\_Agent}) \wedge \ldots \wedge Geq(\mathsf{C}.securityClearance, 3)$$

This request fragment is obtained by forming the conjunction of the subject expression of the rule with those parts of the instantiated object expression that refer to the subject, e.g., through identities. The *instantiated* object expression is the object

expression with the attributes of *object* being instantiated with the concrete object's profile's attributes of the object under access. This instantiation is done at run time based on the accessed object. □

The trick in the above modeling is that the object expression in prediates also referring to the subject is expressed by referring to a variable representing the object's profile that is modeled as an identity and, as usual, variables representing the subject's identities. The object profile variable gets instantiated at evaluation time with the profile of the accessed object and forms additional parts of the data request. For the case that multiple rules apply for a request, all their subject and object expressions are considered in the processing as explained further below. Through the taken modeling approach, the policy elements expressing relations between the requester and the resource are seamlessly integrated with our data model.

The subject-related parts of the object expression need not be specified additionally in the subject expression, but are pulled into the data request at run time, thus simplifying policy authoring.

The processing must concretely perform the following steps in order to handle subject-dependent object expressions: When a specific object is requested, any references to the keyword *object* and attributes thereof of the object expression of the policy are instantiated with the concrete attribute values from the profile of the accessed object. Then the object expression is evaluated and if it does not evaluate to true, the predicates of the object expression that make subject references are composed to the data request to be sent to the other party. Optionally, the idea of sanitizing a data request still applies as proposed by Ardagna et al. [ACDS08] and may remove concrete values in the data request from the object's profile. This allows for optimizing protection for the concrete case, either by better protection of the identity data of the subject or information on access restrictions of the object, depending on what is considered more important.

In terms of privacy, this approach of modeling goes very far with respect to data minimization and anonymity, compared to the original work which needed to always request the attribute value from the subject in case this value was referred to from a predicate in the object expression. In our approach, in many cases, e.g., the agency example above, partial information expressed through a predicate on an attribute is sufficient to compute an authorization decision.

The approach of allowing for subject-dependent object expressions as explained in this sub-section is not new, though was not tuned for privacy in prior work. Our contribution in this area is a solution that offers the power of relating subject data with object profiles in policy rules while allowing one to define policies that either focus on the protection of the subject's data by minimizing disclosure or on the protection of object-related information,

depending on the requirements of the use case. Another related contribution is the integration of the concept into our architecture.

9.7.4.3   Sources of Data

Attributes referred to in a subject expression may be provided by different parties acting as *data providers*. The default case is that the other party in the interaction must fulfill a data request, that is, is the data provider. Though, certain data may be required to be provided from third parties. Let an example for this be the authorization policy of a user protecting her credit card data. For such a policy it can make sense to require, in addition to a proof by the server about some of its attributes, also a reputation score about the service provider, stated by a reputation provider. This can give a user a better capability of assessing a party it is going to reveal data to. The reputation provider then is the third party that provides the reputation score about the service provider without the involvement of the service provider. Let another example, this time for the policy of a service provider, be a service provider whose policy protecting a service is specified to require certain attribut values from a user as well as a minimum credit rating about the identified user from a credit rating company in order to give the user access to the protected service. This example can be realized by exactly the same concept of pulling data from a third party at request time, fully integrated into the authorization subsystem. Such cases have not been explicitly considered in the original authorization model of Ardagna et al. [ACDS08] we build upon, but can enrich the system both from the perspective of a user and a service provider. We think it is particularly important to allow users to pull in information from third parties to better assess other parties they interact with as the obtained information is a foundation for exercising informed consent. All use cases for this feature have in common that data are requested by a party about another party from a third party without the other party being involved in this.

   We note that the approach of pulling information about a user from a third party by a service provider does not fit the user-centric model of data exchange as the third party provides the data without the user necessarily consenting to or even being aware of this and no identity relationship is established for such data on the user's side. Though, the feature is, in the setting of being employed by a user, useful for giving the user more information for assessing a service provider before revealing data to it and thus can enhance user privacy.

Language elements

Executing such a protocol flow requires that a data request is made that, for the third party, unambiguously specifies the subject data about which is being requested. This is done using the request language of Section . We refer to this section also for examples of data requests to third parties. We recall that for this feature the *data provider* must be specified as metadata over

the identity in question, the default provider being the interaction partner. This item of metadata unambiguously specifies who may provide data related to the identity. The granularity of this is clearly the identity being a group of attributes, and thus the natural choice for the atomic unit of attributes that are provided by the same party.

We suggest that the data provider be specified through an attribute of an identity. The possible values of the attribute are interaction_partner or certifier_identifier. The semantics is defined operationally by requesting the data from the interaction partner in the one case and the certifier given for the data in the other case. The constants are interpreted by themselves. The matching of requests containing such a specification of the data provider with data of a party is done as usual and no complication is incurred. Besides this, another possible option for modeling the data provider is to express it as on-formula metadata and have analogous operational semantics, with the constants not being expressed within the data model, but as metadata external to it.

Limitations

We note that pulling data from third parties puts restrictions on the formulae that may be expressed as data requests, e.g., restrictions that statements on identities provided from different parties may not be combined in certain ways as there might not exist protocols for proving that such a statement holds. The conservative approach is that a request formula is crafted as a formula such that it is a conjunction of sub-formulae such that each such sub-formula may be requested from different data sources.

Furthermore, it is important to mention that the party specifying the policy needs to be aware of the data that the third party can provide as well as which data it needs in order to fulfill a request. It is left to a system deployment on how this information is communicated. A simple solution for the user side is to pre-install policies like this for obtaining further information on interaction partners on a user's system or to allow users to obtain them from a trusted policy provider such as a data protection authority. Also, in the console, the feature should be exposed in a specific easily-understandable way in a policy editor, this is not part of this work, though. We provide the technical backbone of the system in this chapter without tackling the user interface.

Regarding the sequence of data exchanges of the party with the interaction partner and the third party or multiple third parties, it is crucial that relevant data is first requested and obtained by the party from the interaction partner as this may be required for specifying the request to the third party. This is implemented in the negotiation protocol of Section 9.9.

9.7.5   Rule Composition

Multiple policy rules may apply to a single resource, e.g., when specifying more concrete access requirements for more concrete object expressions when using a hierarchy for expressing the policy on different levels of abstraction, by simply specifying more than one rule on the resource, or by specifying a rule on an ontology type and another rule on an instance of this type. In such a case of multiple policies being applicable to a concrete resource, all applicable policies apply in a conjunctive way. This means that our policy model requires that in such cases, all applicable rules must evaluate to *grant* in order to give the subject access to the resource. A single *deny* immediately leads to a *deny* response for the overall request.

The reason for policies applying conjunctively and not disjunctively for a class hierarchy is given by the nature of hierarchies. In a hierarchy a class comprises subclasses or instances at each level. A policy specified on a class is intended to hold for all its subclasses or instances. Thus, for a concrete instance, all policies expressed on the class the instance is contained in and all superclasses as well as the policies expressed directly on the instance all equally apply to the instance. This justifies the evaluation semantics for hierarchies. Similar reasoning holds for an ontology type which is the class of its instances using the extensional definition of a class through its instances. For multiple policies applying to a resource, it is a natural way to define the semantics like this. Following this argumentation, any disjunction one wants to specify for the data requirements on the requester must thus be expressed within a single rule as data requirements of rules always compose in a conjunctive way.[19]

Evaluation

A policy evaluation computation is always done with respect to a single resource r; it proceeds as follows: the set of applicable rules is retrieved by evaluating the *object* elements of the authorization rules of the party. For each matching rule, its applicability is checked by evaluating its object expression on the resource. All rules that match again are applicable to the resource. Next, an authorization decision is done for each of those rules. This process uses the dynamic subject profile and the subject profile of the requester and the object profile of r as input. If all rules evaluate to *grant*, access to r is granted; if one or more rules evaluate to *deny*, the access request is denied; if one or more rules evaluate to a grant and the remaining one or more rules evaluate to data requests, a composed data request is created from all the data

---

[19] This model can be extended with the possibility of allowing disjunctions to be specified among rules. In this case, a boolean structure of rules applies to each instance or class. The composition then must conjunct those boolean formulae of rules. The result formula can be represented in disjunctive normal form for a more intuitive representation.

requests. This data request needs to be answered by the subject as a necessary condition for getting access. The evaluation uses the following method for computing the data request.

Let $d_1, \ldots, d_k$ be the data requests represented as formulae of the rules the evaluation of which has yielded data requests. Let the composed data request be defined as the logical conjunction of the individual data requests: $d = d_1 \wedge \cdots \wedge d_k$. Each data request d_j already is built from both the subject and object expressions of the rule as explained further above.

We want to note that the underlying data model we use conveys an important property related to the composition of data requests. The same variables, e.g., for identities, used in different requests d_j refer to the same objects. This allows one, e.g., to specify a request of attribute information via a specific identity variable C in one policy rule and additional attribute information via the same identity by re-using the same variable for referring to the identity in another policy rule. Note that this also holds for conditionally-released identities or opaque identities. As a drawback of this property, different policies are not independent of each other which may make policy authoring harder; though, with tool support this should not be a practical issue.

9.7.6   Associating Policies with Resources

Each authorization policy rule is associated with objects access to which is governed through it. Each *object* element in a policy rule must unambiguously resolve to a set of resources held by the party. Resources of a party are data held by it and services it offers. Data includes both data of which the party is the subject as well as data with other parties as subject. The association is expressed through the *object* element of the policy rule, the resulting set of objects is further constrained through the object expression. The association of policies with resources makes sure that for each resource one can unambiguously obtain all authorization policy rules that apply to the resource. For a concrete implementation, a reverse link from resources to its policies may be maintained or caching or other optimizations may be employed for improved performance of retrieving applicable policies for a resource. Thus, an implementation can deviate from the conceptual thinking explained here for gaining improved performance.

In this subsection we elaborate on the methods and semantics for associating policies with resources in our architecture. This is a crucial functionality of a policy-driven identity management system architecture. This functionality is part of the concrete use of the described authorization system within our architecture and depends strongly on our data representation.

Resources on which policies can be defined are the following: instances and classes where either of those can be data or services. The association is expressed by specifying the resource within the *object* element of the policy rules. The choice of expressing the resource within the policy captures also abstract resources such as ontology types that can stand for multiple concrete

instances of resources. This way, we obtain a uniform way of modeling the associations. It is also the natural way from a policy editing perspective to associate a policy with instance data and classes of data. Each of those needs to be expressable in the object element of the policy rule language.

9.7.6.1   Modeling

Technically, we model all resources using the syntax of our data representation and request language. Also services and their object profiles are modeled using those concepts, alike the modeling of any data. By using the syntax of our data representation to represent policy objects, the already-existing mechanisms of Section 9.3 for computing whether one formula can be used to derive another formula can be applied over the party's data repositories for finding the concrete resources a policy object refers to.

The specification of the target within a policy is a set of tuples (t, c) with t being the target expressed in our data model and c being a condition expressed on the target to specify further restrictions. We may notationally only present a single element, in case only t, but no c, is given. The target corresponds to all data items d such that for each of those items the following holds: $t \wedge c \vdash_{\mathcal{O}} d$ over an ontology $\mathcal{O}$. Note that all free variables of t and c need to be instantiated accordingly. The target instance is $d' \preceq d$ with the sequent $t \vdash_{\mathcal{O}} d$ holding. This informally means that only t is "matched" with the target while c further restricts the results of this matching.

A target can contain variables and in the case of the variables being free variables and no environment being given that instantiates them, the target is a class of formulae. In case of an environment being given, e.g., in a data formula that responds to a request, the policy applies to the instantiated variables, that is, includes the values given in the environment. This avoids any complication in the meaning of the policy association by keeping the predicate as smallest atom to which policies are associated.

The use of the data model for specifying policy targets immediately transfers the expressivity of the data model to the policy target specification. Particularly reasoning capabilities are available as well which allows, for example, for expressing hierarchies over ontology types. For a data statement, all free variables must be instantiated as usual through an environment and a policy applying to a variable means it applies to the value it has assigned. For a data request, free variables may not be instantiated to represent that they need to be provided by a formula fulfilling the request. Such a formula stands for the class of data statements that fulfill it. Analogously, the policy expressed on the formula applies to any data formula that fulfills the request.

We next present the different ways of associating policies with resources. Basically, there are two different ways of associating policies with data: on the level of instances and of classes. Instances can be be instances of data or services, and likewise, classes can be classes of data and services as well. The idea behind allowing both is that specifying a policy on classes allows for

leveraging the concept of inheritance of policies among classes in a data type hierarchy, analogous to the inheritance concept in object-oriented programming languages.

Instances

Instances can be instances of data or services. Examples for instance data are an attribute with its value, a complete identity, that is, all its attributes, or a formula or sub-formula of a formula. A (sub-)formula can particularly be one describing an identity relationship. In this case, the authorization policy drives negotiation protocols of the party with other parties, see Section 9.9 for details. A formula may also contain disjunctions or just comprise a single predicate, e.g., one that expresses an attribute-value pair. An example for a service is a concrete instance of a service referred to by an identifier. Any of those data or service instances can be specified in object expressions.

Example (Object specification for instance data): ϕ: Name of a (sub-)formula of arbitrary complexity. In the simplest case a formula is a predicate expressing an attribute-value pair, that is, a simple instance data item, e.g., Eq(id4567.*lastname*, Doe).

id4567.*lastname*: An attribute value. This is a shorthand form for a formula representing the attribute through a predicate as shown above.

Gt(id4569.*reputationscore*, 6): This expresses partial information about an attribute expressed through a predicate over it. This is an interesting case in the light of specifying less strict authorization policies on partial information of an attribute than on its concrete value.

id4160: An identity. It means all attributes of the identity with their values.

profile5640: Profile profile5640 held by the party. This includes all the profile's attribute information. The profile is modeled as an identity.

profile5640.*lastname*: As is the case for any identity, more specific profile data can be addressed, e.g., a single attribute as shown in the example.   □

Services are technically represented exactly like data having an associated identity for each service the party offers. The identity is used to express the *object profile* of the service. We adopt the convention that the keyword (attribute) *service* of a service identity refers to the service itself to distinguish it from its object profile and to allow for restricting access to the profile while giving access only to the service. The service can be provided in any way, independent of our authorization architecture. The authorization architecture is only responsible for decisions on access to the service.

Example (Object specification for services instances):
service68: This represents the service service68 of the party including its object profile, all modeled as as one identity.

service68.*service*: This represents only the service **service68** itself, without the data of its object profile. In the common case, it is sufficient to give requesters access to only the service.   □

Classes

A class is a set of classes or instances and can be specified through a formula comprising free variables. An instance of a class is a formula such that it can be derived in our data model's calculus using an appropriate instantiation of the free variables. The most general case of a class is a class of formulae. When restricting this, we can obtain classes of predicates of a certain form. An even more concrete case is the one of ontology types.

Ontology types are a specific and practically important case of classes in that an ontology type represents all attribute values of its type through a specific form of predicates. An ontology type is specified by a formula $Eq(\mathsf{C}.o, \mathsf{V})$ where C is a free variable representing an identity, o is an ontology type for an attribute, and V is a free variable representing the attribute's value. This formula refers to the set of all data formulae of the party with the variables being replaced by concrete terms like in $\phi = Eq(\mathsf{c}.o, \mathsf{Doe})$. Each such formula represents, because of using the *Eq*-predicate, an attribute-value pair related to an identity with the ontology type o, as stored in one of the data formulae of the party. It does not capture other predicates like greater than which is intentional as this should only capture attribute-value pairs.

Using the standard reasoning capabilities of our data model, it is possible to express hierarchies over ontology types in an ontology and refer to classes in such hierarchies as policy targets. Expressing an authorization policy on a class in a hierarchy can, for example, be used to express a single policy for all certification metadata attributes of all identities of the party's identity relationships.

Example (Object specification for an ontology type):
$Eq(\mathsf{C}.lastname, \mathsf{Lastname})$: This is equal to the set of all *lastname* attributes of all formulae held by the party.

$(t, c) = (Eq(\mathsf{C}.X, \mathsf{Y}), IsA(X, \mathsf{O}))$: This refers to all equal-predicates referring to an attribute X of any identity C where this attribute is of ontology type O and has value Y.   □

Expressing authorization policies on ontology types is the preferable approach if multiple instances of data are available of the same ontology type, as for example for each credit card number attribute of the different credit cards of a user or all *lastname* attributes of a service provider's customers, and if the same policy rule is to be expressed on all those instances. This does not prevent additional policy rules to be associated with individual instances, which can, for example, be used by a service provider to specify additional protection

requirements for data agreed with a specific customer. For a service provider, it is particularly important for reasons of simplicity of policy management to be able to associate authorization policies with ontology types.

The idea for representing ontology types may be used for referring to any other predicates with useful applications in practice. For example, a policy can be associated to all data statements that reveal that the *validUntil* attribute of an identity is greater than now without giving its value.

Example (Object specification for predicates):

The formula $\phi = \mathsf{Gt}(\mathsf{C}.validUntil, \mathsf{Now})$ represents all instances of the greater than predicate expressed on the *validUntil* attribute of any identity C on the variable Now that gets instantiated with the current time. Now is a variable that gets instantiated with the concrete value at the time of evaluation of the formula. Such a formula can be used to define a policy on any statement that a certificate is valid from a temporal perspective. $\qquad\square$

The ideas presented above for representing classes of predicates can be applied to expressing more general formulae using the same approach of representing identities and constants as variables. This is an extremely general way of associating policies with resources and may have certain use cases, e.g., to associate a policy with a widely-used disjunctive statement. Though, we would like to note that the most prominent use of classes for policy targets are ontology types, with the extension of referring to concepts in an ontology, such as hierarchies.

Expressing an authorization policy on a more complex formula can be useful, for example, for commonly-used parts of formulae, e.g., the statement that the party is from a European Union country. This would be realized by a disjunction on the citizenship attribute of a party over all EU countries. Such parts of formulae may be agreed on through ontologies to which parties refer when using them.

For all cases of expressing classes of data as policy target, the meaning is that any instance data formula that fulfills the target definition is an actual target instance.

We note that the mentioned concepts for associating policies with resources apply not only to authorization policies, but also to data handling policies which are discussed in Section 9.8. We also note that a data request in an authorization policy is nothing else than a specification of a class of instances, any of which will fulfill this request. Furthermore, we note that the presented concept of associating policies with resources can be simplified in a specific instance of our architecture in a concrete system, e.g., for the reason of simplicity of an implementation. The PRIME Prototype has taken simplifications to show the basic ideas of the authorization system while not implementing all its features.

9.7.6.2 Discussion

The overarching design principle for the policy association is to specify policies only on identity data itself and not on the cryptographic tokens that are used to execute protocols related to such identity data, e.g., private certificates. This principle of considering the data as the primary elements of interest, and not associated cryptographic tokens, reflects basic identity management needs and is crucial for the architecture. Once an authorization decision for releasing data has been made on the identity data, cryptographic tokens are only used as governed through such a decision to perform a data release protocol based on private certificates. Defining an authorization policy on a token like a private certificate would not be a clean solution for the following reasons: A private certificate is only one specific tool used of releasing certified data, while our data-based approach is generic and applicable to a wide range of data release protocols. That is, the policy association would have to be done separately for all data release protocols to be supported. Our approach, on the contrary, uses our data model as technology-independent data representation of identity relationships on which all decisions are made. Only for executing the actual release protocol, we revert to the concrete technology.

Our approach to the granularity of defining policies on data allows for great flexibility and expressiveness in the policy definition: objects of any granularity—from partial information on attributes to data types—can be addressed. This particularly allows for having policies apply at different levels of granularity for a single item of data. Take as an example authorization policies to be defined on individual attributes such as the first name or postal code and additional, stricter, authorization policies being defined on formulae comprising multiple of the attributes, such as one revealing both the first name and postal code. Stricter policies governing the release of such combinations of attribute values or more general predicates can be useful in cases where a user's anonymity set may be too small for certain attribute combinations.

Note that a useful extension for all policy associations using classes is that the explained semantics be applicable to a specific kind of data of a party only, e.g., all data profiles or all identity relationships. The restriction on the kinds of data is useful, for example, for allowing a party to specify different policies for their identity relationships and data of other parties they hold which is a natural requirement.

For most practical use cases, applicable policies need to be found based on a resource r, such as a data item, thus the above needs conceptually to be executed for all policies and all policies that yield the data item in their concrete set of target objects are applicable to the data item. Finding the applicable policies for a resource r proceeds as follows: for each policy rule p, take its *object* element and perform a check whether the resource is contained in the set R_p of resources specified by it, that is, whether $r \in R_p$. If this holds, add them to the set P'_r of policies. Next, check for each $p \in P'_r$ whether the object expression of p matches the resource. This involves the object

profile and the static and dynamic subject profiles. If it matches, add p to the set P_r, the set of policies applicable to resource r. This processing is useful to formally define the semantics of the policy targets, though not to act as specification of the concrete implementation. A practical implementation should not implement this semantics in a straightforward way, but optimize it to obtain practical performance figures. Possible optimizations are indexing of the data as well as performing precomputations, and combinations thereof. This is particularly true for systems that handle large amounts of data, e.g., systems of service providers.

9.7.7   Architectural Integration

As part of the overall architecture, the authorization functionality as described above is realized by a distinct component, the *authorization* component. The component is stateless and implements an authorization policy evaluation engine for our authorization language.

An authorization request issued to this component is a tuple specifying the following: the resource that is to be accessed; the action to be performed; data specifying the requester; the purposes of the request. One authorization request may ask for access to a single resource instance as a request cannot have a class of resources as target. Resources are addressed by means of data (resource) requests expressed in our data model as explained in Section 9.3.

In case multiple resources are to be accessed by a requester or component, individual requests must be issued to the component and the responses handled by other components. This is done by the negotiation component which drives mutual data exchange between two interacting parties as explained in Section 9.9. Though, authorization queries are done independently of negotiations, e.g, when an authorization decision for a local access to data must be computed.

The authorization component needs to access the policy management component for obtaining policies for use in the evaluation as well as the data store in order to access subject and object profiles at evaluation time. Any required state is passed to the component as input for an evaluation request.

The mapping, also of the policies, from a data request to a concrete data item to be released to the other party in an interaction is done within the process of a negotiation as explained in Section 9.9. This mapping may require input by the human for a party being a user in the case that the mapping is not uniquely determined, e.g., the party has multiple identifier and identity relationships that can be equally used for fulfilling the request. Based on a choice by the party on the concrete data to release, the applicable authorization policies can be computed which is also done within the negotiation protocol.

9.7.7.1 Data Integration

For an integration of the authorization policy evaluation component with the architecture, the policy evaluation engine must be able to access data about the subject (access requester) and about the object (resource) being accessed.

Subject Data

Regarding data about the subject, a hybrid approach is taken in our authorization architecture by allowing data from the *dynamic subject profile*, which is the data provided by the other party or third parties during the ongoing session, to be used, as well as data from the (static) *subject profile*, which comprises all formulae stored in the profile data record about the requester. From a technical perspective, both those data sources are realized in the same way: They comprise a set of formulae about the subject, expressed in our data model. For a policy evaluation, the conjunction of those formulae represents the data known about the subject. Conceptually, there are differences, though: For retrieving and utilizing the static subject profile, the requester needs to be identified, at least with respect to a pseudonym it is known under at the party. Thus this approach is not applicable to anonymous interactions – at least pseudonymity is required. Furthermore, the data in the subject profile are not necessarily up to date as they may have been obtained in an earlier session. Thus, certain attribute information may be re-requested from the requester in the current session, even though they are already available in the subject profile. The dynamic subject context is populated with the data received through all the data release protocols during the ongoing session. It is particularly suitable for realizing authorization in anonymous interactions. In such, it is the only source of authentication information available about the requester for computing authorization decisions. It also holds a possible identifier of the requester (subject) that can be used to retrieve the static subject profile about the requester.

As the dynamic subject profile and static subject profile both contain formulae used as input to the policy evaluation process, this approach can be seen as an extension of the model underlying the work of Ardagna et al. [ACDS08] as the latter is restricted to attribute-value pairs for representing the subject data which implies some restrictions on the expressiveness of their system in terms of data minimization.

Object Data

The *object profile* is modeled in a different way than the subject profile in that the object profile is represented by a single identity which comprises all attribute values as data held about the object (resource) the profile belongs to instead of modeling the data with formulae. The restriction to a single identity allows for the instantiation of the *object* variable with the identity of the profile. We take this simplification of specifying the attributes directly

without reverting to a formula as we do not see a strong need of the additional expressivity for the object profile. This approach allows us to relate an object profile with subject data in a simple way as outlined further above. As an example for an object profile, consider as object a movie that is offered for download by the service provider. The object profile of the movie then captures, for example, its title, genre, age requirements, and director. All of those profile attributes of the movie can be used to express authorization constraints in our policy language on the actual resource which is the movie.

9.8   Data Handling Policies

As an orthogonal feature to the privacy-enhancing release of data, our privacy architecture integrates with concepts of *data handling policies*. Data handling policies specify how the data they are associated with are to be handled by the data's recipients once released. In this section, we focus on a conceptual discussion on data handling policies and their integration into the architecture. We do not propose a concrete policy language or concrete extensions to related work on such languages, thus our work on data handling policies can be seen as a framework for integrating data handling policies into our or similar identity management architectures. For a concrete system based on our architecture, a concrete language still needs to be chosen for this purpose. This allows one to retain flexibility in the concrete data handling policy language to use while still presenting the basic architectural ideas and limitations. We provide a rough formalization of the main aspects of the model underying our framework to make the thoughts clear and to use it for the discussion of policy negotiation and related parts of the negotiation protocol of Section 9.9. We do not provide a rigorous formalization of our approach to data handling policies.

9.8.1   Model

The setting we consider is the usual one of multiple parties, each of which may interact with each other. We only consider interactions between two parties at the time (within a protocol). Concretely, a *data provider* (e.g., the subject of the data or a user) interacts with a *data recipient* (e.g., a service provider) and needs to release data to the data recipient. The data provider releases data to the data recipient, typically data that have been requested by the recipient earlier in the interaction. The recipient may need to release parts of the data to other data recipients (third parties), for example, for providing a service, as is the case for the third party being a shipping company intended to deliver ordered merchandise to the customer, or for other purposes, such as statistics, direct marketing, or receiving product updates. Note that data provider and data recipient are again roles that characterize parties for parts of an interaction.

Each party can have data handling policies specified for each data item or class of data items represented at the party, regardless of the party. Being more specific, the data provider can, for each data item that can potentially be released, have data handling policies specified. The data recipient can have data handling policies specified for its data request parts of its authorization policies as those request parts determine the data to be provided by a data provider. Those policies of the data provider and the data recipient are the basis for an agreement process on the actual policies to be enforced on data items to be released. This agreement process is the *policy negotiation*. The resulting agreed policies will be maintained by the data provider in the data track entry related to the data release as well as by the data recipient in the corresponding profile data records representing the released data. Note that within a single session, the roles of data provider and data recipient can be swapped multiple times for a mutual release of data. This is exactly the case in the negotiation protocol of Section 9.9.

Typical practical settings for a policy negotiation are the following: The data provider is a user who is also the data subject; she provides data to a service provider. The data provider is a service provider who has previously received data from a user; the data is then released to another data recipient. The case of the data provider being a service provider and providing data about itself to a user as data recipient is less interesting because a service provider will usually not impose usage restrictions on data related to itself. In all those cases, the data provider (user or service provider) has a data handling policy specified on instance data (identity relationship or profile data) that is enforced when releasing data.

Definition (Data handling rule): A *data handling rule u* is a statement in a formal language that specifies data handling provisions. A rule can impose restrictions or rights regarding the handling of data the policy it is contained in applies to. A *data handling policy p* comprises a set of rules $u_1, \ldots, u_{k_p}$.   □

A rule/policy is associated with resources using the same means as applied to authorization rules discussed in Section 9.7, that is, the rule expresses the data it applies to.

Definition (Subsumption): A rule $u_i \preceq u_j$, in words u_i *subsumes* u_j, holds if and only if the following holds: If u_i is fulfilled, u_j is fulfilled as well. Analogously, we define $u_j \succeq u_i$, in words u_j *is subsumed by* u_i.   □

We cannot give a rigorous account of what it means for a rule to be fulfilled as we do not fix the language as we want to keep the architecture open for different rule languages. We will base some of our definitions on the concept of a rule being fulfilled.

To give some intuition on what rule subsumption means, consider the following: A rule u_i subsuming another rule u_j means that whenever u_i is *enforced*, also rule u_j is enforced. Hereby, we assume that the concepts of fulfilling and enforcing a rule are equivalent. One can also see this relation as u_i being *stricter* than u_j. For example, a rule that states that data are deleted after 3 to 4 months of retention subsumes a rule that states that data are deleted within 0 to 5 months; whenever the first rule is fulfilled, the second one is also fulfilled.

Definition (Proper subsumption): Similar to the above, we define the following: A rule $u_i \prec u_j$, in words u_i *properly subsumes* u_j, holds if and only if $u_i \preceq u_j$ and $u_i \neq u_j$. Analogously, we define $u_j \succ u_i$, in words u_j *is properly subsumed by* u_i. □

Definition (Intersection): Definition: The *intersection* $u_i \cap u_j$ between two rules u_i and u_j is the rule u' such that $u' \preceq u_i$ and $u' \preceq u_j$ and there is no rule $u'' \neq u'$ such that $u'' \preceq u_i$ and $u'' \preceq u_j$ and $u' \preceq u''$. □

Intuitively, this means that u' is the least restrictive rule, that is, the rule that is least constraining, that subsumes both intersected rules. For the example of data deletion rules further above where one rule specifies deletion of data within 0 to 5 months and the other between 3 to 4 months, the intersection of those is a rule stating that data are deleted after 3 to 4 months, thus fulfilling both rules of the intersection. A precise definition must take the concrete rule language and deontic meaning of the rules into account, that is, for example, whether a rule has the meaning of "may be done" or "must be done".

Subsumption is defined on policies as well. Intuitively, a data handling policy p_1 *subsumes* a data handling policy p_2 ($p_1 \preceq p_2$) if and only if the following holds: Whenever p_1 is fulfilled, also p_2 is fulfilled. Subsumption on policies generalizes the concept for rules to policies, that is, sets of rules, with the concepts remaining the same and no details being given due to no concrete rule language being defined. Analogously to rules, the relation $p_2 \succeq p_1$ is specified as well.

In most practical scenarios multiple policies are simultaneously communicated and agreed on for multiple resources during a negotiation protocol. We refer to multiple such data handling policies as *structure of policies*. We require that the representation of data ϕ and an associated structure of policies π unambiguously allow for determining the association between items of data or request and policies. This is practically done through our means of associating policies with data as explained in Section 9.7, now applied to data handling policies.

Definition (Subsumption on policy structures on data): Let π_1 and π_2 be structures of data handling policies. Let π_1 be associated with data represented through formula ϕ_1 and let π_2 be associated with data represented through formula ϕ_2. The *subsumption* relation $\pi_1 \preceq \pi_2$ between structures of policies on data is defined as follows: $\pi_1 \preceq \pi_2$ if and only if $p_1 \preceq p_2$ for each pair (p_1, p_2) with $p_1 \in \pi_1$ and $p_2 \in \pi_2$ and p_1 and p_2 applying to corresponding data items. $\qquad\square$

This definition transfers the above definition for policies to structures of policies and is easier to work with in a practical setting of negotiation between two parties. Note that this definition is based on the definition of the subsumption relation for policies without having given a concrete policy or rule language and semantics.

The above definition for intersection of rules carries over to policies and structures of policies. Those operations are relevant for the implementation of negotiation of policies between two parties and used in Section 9.9 on negotiation. Again, we cannot give a precise definition as we do not fix the rule language.

The definitions and discussions so far have covered data handling while abstracting the concrete rule and policy language. The basic meaning of a data handling rule is defined through the vocabulary of the rule language and related ontologies. Further meaning that governs the processing is defined through operational semantics in the way policies are used by a party. A policy can be either a *requirement* of a party or a *proposal*. A requirement is a policy that *must be enforced* by the party on the data it is associated with. A proposal is a policy for which a party proposes that it *will enforce* the policy on data received from other parties, that is, there is no concrete data available at the party that is the target of this policy. For a specific data item to be released by a data provider to a data recipient, the two parties agree on a policy that subsumes the requirement of the data provider as well as the proposal of the data recipient. This new policy becomes the requirement of the data recipient for this data item once the data item has been released by the data provider to the data recipient. A requirement can be specified on both instance data and classes of data such as ontology types or requests. A proposal is specified on classes of data only as the instances are not known at the time of defining the policy. The same language can, but need not, be used to express both a requirement and a proposal – the semantics can be defined operationally by using a policy as a requirement or proposal. Such operational semantics is implemented through the policy negotiation process and the appropriate use of the policies in it as well as the later enforcement of the agreed policies.

Both rules by a party that restrict a party in handling data and rules that grant it rights can be handled in the same way in our model. An example for the further is a rule requiring data deletion within 3 to 4 months of having received the data, for the latter a rule allowing to use the data for marketing

purposes under defined restrictions. For both cases, the data provider can specify a rule that the data recipient must follow – such a rule may mandate an action or allow one. Analogously, the data recipient can specify a rule with the meaning that it will perform an action or wants to do so. For a concrete rule language we need to be able to express the related meaning of the different intentions of the parties, whether they require or allow actions or will or want to do something and the language must allow for implementation of the required relations and operations such as subsumption or intersection on rules.

9.8.2   Association of Policies with Data

Data handling policies can, much like authorization policies, be defined on instance data items or classes of data in a spectrum of granularities ranging from atomic items of instance data, over parts of formulae or complete formulae to classes of atomic items or classes of sub-formulae or formulae. See Section 9.7 on authorization policies on details how the association of policies and data is expressed in the definition of the policies. The same mechanisms and granularity of data as mentioned there apply also to the association of data handling policies with data. We assume that the data handling policy language provides for a language element for expressing the policy target, like our authorization policy language.

Every party can associate data handling policies with classes of data by expressing the class through a formula in the policy target, or with instance data by expressing the instance in the policy target. A specific case of associating data handling policies with classes of data is where the class is a subject expression of an authorization policy. Figure 9.5 illustrates the association of both authorization and data handling policies with data instances and classes. Each data handling rule on instance data or classes of data can be a data handling proposal or a data handling rule that has been agreed with a data provider and that must be enforced. A data handling rule on the subject expression of an authorization policy is always a proposal of data handling. A service provider will typically specify its data handling policy proposals on classes of data to be requested as part of its authorization policies. This can be done by associating the data handling policies with the request parts of authorization policies that protect its resources. Those request parts are expressed in the data request language and may comprise variables for identities and attributes. A party like an end user will specify her data handling policies on ontology types whenever possible. For example, a user can specify policies on the ontology type of credit card data, thus covering all instances of credit card numbers, expiration dates and so on, as the ontology relates the concepts. A user can then specify a more restrictive policy for their credit card with high spending limit by associating the according policy with the instance data representing this card. This data item would then inherit the policy defined on the class of credit cards, thus both policies apply to it.
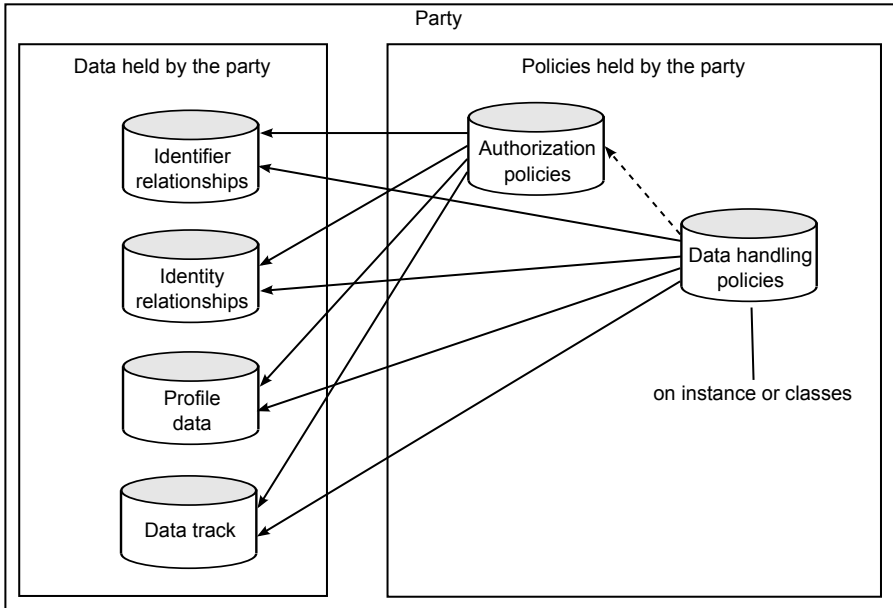
Authorization and data handling policies expressed

⟵────────────   on instance data or classes

⟵- - - - - - - - - -   on data requests (classes)

Fig. 9.5 Association of data with data handling policies

Note that the use cases of our architecture require data handling policies to be specified on data and data requests, but not on services or data provided as a service, such as download of music. Defining data handling policies, in this case better called service handling policies, on services would bring us into the area of digital rights management, another aspect of *usage control*. We will not further consider this in the architecture, but note that conceptually the agreement of such policies is integrated into the architecture, the enforcement is a different and quite difficult problem.

A specific particularity that deserves additional consideration are data requests that contain *disjunctions* as allowed by the data model. As those may be beneficial for data minimization and thus user privacy, we will discuss the specifics of disjunctions in some detail. A data request with at least one disjunction can be fulfilled by a subject either by proof, that is, by performing a proof that an instantiation of the whole request holds, e.g., using private certificate protocols, or by choice, which means by choosing a specific "branch" of each disjunction and providing data and proof about the data. In certain cases, the service provider might want to restrict in which way the subject can fulfill the request. Regarding data handling policies, disjunctions

in a request complicate the situation: In case the service provider restricts that either specific parts are released and proved to hold by the subject or the full request formula must be proved, data handling policies are associated with each "branch" and its items of each disjunction to cover for the case of a branch being selected, or with the full request formula to cover for the case of the full formula being proven to hold. In case the service provider does not impose restrictions on how the formula is proven, policies need to be provided for both cases. No specific language element is required for this, but the semantics is expressed by associating a policy with the correct element: either a subtree with a disjunction statement as root, or elements within a disjunction statement. Once the subject has chosen how to fulfill the formula, the relevant policies are used for an agreement on the actual policies to be enforced. In a negotiation, we always have a concrete data statement as input that fulfills the request. We note that for data containing disjunctions, data handling policies are often less crucial than for other requests as the disjunction leads in many cases to a substantial increase of the anonymity set the subject is in which is also the main reason why disjunctions have been introduced into our data model and our architecture for releasing data statements.

Example (Data request for policy association): The following shows a fraction of a request formula ϕ for data that is part of an authorization policy of a service provider. Data handling policies can be associated with the individual predicates requesting statements on attributes related to the requester with the meaning that the predicates with the free variables being instantiated in the subject's response will be subject to this policy. A data handling rule can refer to any of the data items in this formula, that is, individual predicates or any other sub-formula through the definition of the target element of the rule as described for authorization policies in Section 9.7.

$$\phi = Eq(\mathsf{C}.firstname, \mathsf{Firstname}) \wedge Geq(\mathsf{C}.monthlysalary, 3500) \wedge$$
$$Eq(\mathsf{C}.monthlysalarycurrency, \mathsf{EUR}) \wedge Eq(\mathsf{C}.type, \mathsf{Bank\_Statement}) \wedge \dots$$

The predicates in a response to the example request correspond to attribute values with identity terms that are chosen by the data provider, in the example equivalent to simple type-value pairs together with the identity term because of the $Eq()$-predicate being used. A policy on the whole formula ϕ means that it applies to all of its $\wedge$-connected predicates, also those representing certification metadata or other metadata. The meaning for sub-formulae is analogous. $\qquad\square$

The metadata in a formula are much less crucial than a party's identity data in terms of data protection in the standard use cases. Though, in specific use cases, the fact of having an identity relationship of a specific type from a certain certifier may already be valuable information. For this reason, it is preferable to be able to protect those metadata equally to data. This is the case in our model thanks to the uniform modeling of data and metadata in the data formula.

When data is being requested or provided, that is, communicated over a network, data and policies form a unit and are formally associated with each other on the wire in the same way as they are in the (logical) representation at the party. The difference is that on the wire, data and policies are often contained within a single message while at a party the two are stored in separate repositories. The formal association remains exactly the same in either case, leading to a simple model with a uniform data and policy representation and meaning.

9.8.3   Policy Negotiation

An integral part of an exchange of data between a data provider and a data recipient in our setting is the agreement on the data handling policies to be applied to those data – the *policy negotiation*.

Policy negotiation happens as a part of the process when two parties engage in a negotiation protocol as described in Section 9.9 to agree on the data handling policies to be applied to released data. We denote the participating parties by their roles in terms of data exchange: *data provider* and *data recipient*. We assume that the data provider has received a data request from the data recipient, e.g., based on one or more authorization policies of the recipient. We assume also that the data provider has decided with which data to fulfill the request.[20] Let the data be denoted $d_{\mathcal{P}}$. Let us first, for simplicity reasons, assume that a single item of data is concerned, e.g., an $Eq(\_,\_)$-predicate expressing an attribute value. The approach then generalizes to structures of policies over formulae of data or requests taking into account the discussion further above on the properties of structures of policies associated with data. We assume also that both parties have a single data handling policy associated with the data and the request, respectively. The approach can then be generalized to structures of policies on formulae for the practical case of a policy negotiation.

The data $d_{\mathcal{P}}$ held by the data provider $\mathcal{P}$ has a set of data handling rules, that is, a data handling policy $p_{\mathcal{P}}$, associated with it. This policy has, as it is defined on instance data, the semantics that it must be enforced on the data by the data provider and future recipients, that is, it is a requirement. The data recipient $\mathcal{R}$ has a data handling policy $p_{\mathcal{R}}$ associated with the data request. This policy has the semantics that the data recipient will enforce it once it receives data satisfying this request, that is, it is a proposal by the data recipient $\mathcal{R}$. The policy negotiation protocol has as inputs the policy $p_{\mathcal{P}}$ by $\mathcal{P}$ and the policy $p_{\mathcal{R}}$ by $\mathcal{R}$. The output is a policy $p' = p_{\mathcal{P}} \cap p_{\mathcal{R}}$ with the

---

[20] Note that the decision on which data to release in a negotiation can be intertwined with the data handling policy negotiation in order to achieve best-possible decisions in terms of data privacy or to find matching policies. We use a simplifying assumption in the explanation of policy negotiation of those decisions being handled sequentially with the option to revisit a decision on the data if no matching policy can be found.

properties $p' \preceq p_{\mathcal{P}}$ and $p' \preceq p_{\mathcal{R}}$, that is, the agreed policy p' subsumes both parties' policies provided as input. Figure 9.6 illustrates the situation at both parties before a policy negotiation and data exchange, the negotiation, and the situation after a successful policy negotiation and data exchange. Elements that we have typeset in bold are the result of the successfully executed policy negotiation.
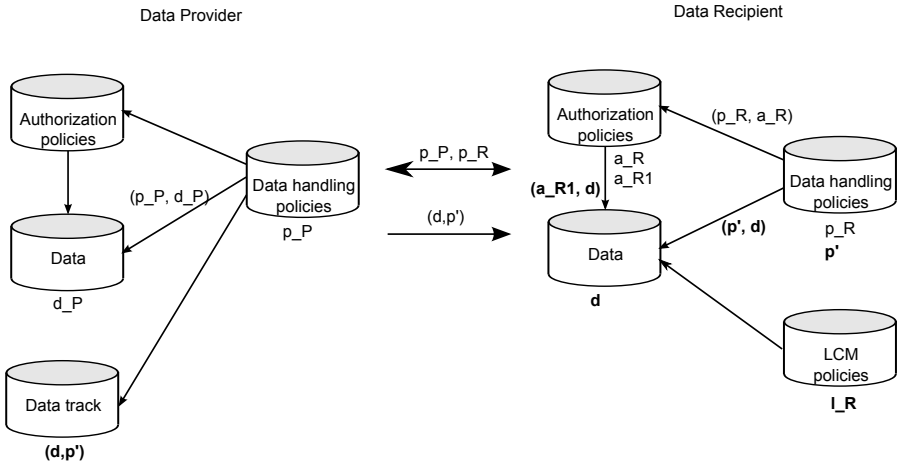
Fig. 9.6 Policy negotiation and data release

There is a variety of possibilities how the negotiation of policies can be technically implemented. The best approach for confidentiality of the policies is a cryptographic multi-party protocol, e.g., based on the evaluation of circuits in zero knowledge. Although such solutions are efficient, that is, of polynomial space and time complexity, they are not practical. A more practical solution from the perspective of computation is a protocol in which the data recipient provides its proposed data handling policy to the data provider, e.g., together with the data request. Then, as the main step of the negotiation, the data provider alone computes a policy p' such that $p' \preceq p_{\mathcal{P}}$ and $p' \preceq p_{\mathcal{R}}$. The simplest method to do this is to compute the *intersection* $p_{\mathcal{P}} \cap p_{\mathcal{R}}$ of policies $p_{\mathcal{P}}$ and $p_{\mathcal{R}}$. Once such a policy is found, the data provider can release the data $d_{\mathcal{P}}$ subject to the policy p'. The recipient checks whether the tuple $(d_{\mathcal{P}}, p')$ is well formed, particularly that the policy target of p' is the data item $d_{\mathcal{P}}$ and $p' \preceq p_{\mathcal{R}}$. If this holds, the policy negotiation terminates with success and the recipient has accepted to handle the received data under the agreed policy. In case the sent policy p' does not subsume $p_{\mathcal{R}}$ or does not apply to $d_{\mathcal{P}}$, the negotiation terminates with failure.

Considering the concrete case of a user being data provider and a service provider being data recipient, the computation of the policy p' in a negotiation

will proceed such that the policy is computed by the user in a way that it reflects the strongest policy in terms of her privacy. Such a policy can be more "restrictive" than the intersection between the user's and service provider's policy. For anonymous interactions, it must be taken care of that the concrete policy computation algorithm does not lead to a gross reduction of the anonymity set the user resides in, e.g., by very specific values of parameters, such as the data retention time, being used.

In the above simple approach to policy negotiation, it is guaranteed that the resulting policy, if it can be found, fulfills both parties' policies – this being the basic success condition of a policy negotiation. The drawbacks of this simple approach is less flexibility than one could obtain by applying a more interactive negotiation protocol with both parties making choices as well as no means of resolving a situation where no policy p' can be found that subsumes both parties' policies. A solution to finding no suitable policy is creating an *exception* at party $\mathcal{P}$. An exception can be realized by the party defining an additional policy applicable to the data item for which an exception is needed and the ongoing session and the computations being based on the new resulting policy set.

We note that the architecture and underlying policy model are open for extensions towards more elaborate negotiation protocols than explained above, e.g., ones using multiple interaction rounds and having different tradeoffs in terms of how much of the policy of the interaction partners is revealed. Such protocols are more suitable than the one described above, for example, for the interaction between two users where both parties have equal protection requirements regarding protection of their privacy. This is a quite different case compared to the case of a user and service provider interoperating as there the user clearly is the more protectworthy party in terms of privacy. Such more elaborate scenarios may deviate from our model in that the parties do not have policies as input, but rather have the constraints of the negotiation specified in a more abstract way. A negotiation can then proceed differently and must fulfill the constraints of the parties in order to be successful.

Optionally, additional protocol steps may be included in a policy negotiation protocol that allow the data provider to obtain non-repudiable digital evidence on the acceptance of the policy from the data recipient. This could be in the form of a digital signature on the data-policy tuple (d, p') that the data provider can retain should there be an issue with data handling through the data recipient or transitive recipients later on. Such evidence can be modeled as part of the metadata in the data track entry resulting from the release of $d_{\mathcal{P}}$ to $\mathcal{R}$. This may put users into a stronger position when it comes to a legal dispute regarding improper handling of a user's data.

Once a policy negotiation and data exchange have been successfully concluded, both parties update their states as follows. The data provider creates a data track entry (or extends an existing one) reflecting the release of the data item $d_{\mathcal{P}}$ under data handling policy p' to party $\mathcal{R}$. The entry may get new authorization policies associated to protect its use, e.g., allowing only

for local access by the party and its identity management system. Existing authorization policies may already apply to data track entries through target specification on the class level. The recipient stores the received data in its profile data store. Authorization policies may be defined or may already apply on a level of classes, e.g., on ontology types or all profile data entries. Authorization policies may also be derived from the data handling policies and applied to the data item d as outlined below. The recipient stores the negotiated data handling policy p' in the policy repository. The policy has the data item as target already which has been validated on reception of the tuple (d, p').

As next step, the party needs to *operationalize* the data handling policies on the received data. The part of the data handling policies that is related to authorization is mapped into concrete authorization policies that are associated with the received data. This part is then enforced through the party's authorization system. The part of the policy that is related to life-cycle data management is mapped into policies that are associated with the data and pushed to the data LCM component. This part of the policy is then enforced through the data LCM component which monitors events and executes actions based on those, e.g., the time-driven deletion of data or notification of the user about policy enforcement related to her data. See Chapters 12 and 15 for details on the concrete enforcement mechanism and the component used in the PRIME prototypes. We stress that the data handling policy agreed with or obtained from the data provider remains the policy, possibly with additional restrictions imposed by the party itself, that is used in policy negotiations with other parties regarding the data item.

Optionally, the model may be extended with the data provider also providing its policy $p_{\mathcal{P}}$ to the data recipient and the data recipient further using this policy for agreeing policies other data recipients need to enforce on the associated data item when further disclosing it. The advantage of this is that the policy $p_{\mathcal{P}}$ is not constrained further by $p_{\mathcal{R}}$'s policy as is the case for p' as explained above. Thus, it is more likely, that a non-empty intersection is found with the policies of other data recipients. The problem in terms of privacy with this approach is that a policy may identify a data provider or allow linking multiple transactions of it. This is not a problem in settings where the data provider is identified in which cases data handling is most important. Metadata on the policy and p' can be used to express how the policies need to be handled by the party. We note that whether a party uses this extension depends mainly on non-technical factors such as regulations and internal policy of the data recipient.

9.8.4 Concrete Realization in the PRIME Prototype

The concrete approach taken within the PRIME prototype and described in later chapters of this book is a simplified one that can be seen as a specialization of our general ideas. To summarize the concrete approach taken, the

data handling policies are strongly integrated with the authorization policies in terms of language and processing. The data handling policies are specified by the data recipient, the data provider may optionally specify its preferences as rules. The negotiation is performed by the data provider instantiating uninstantiated parts of the proposed policies of the data recipient by making choices, e.g., to opt in for the use of the data for direct marketing purposes. The agreed policies are then added to the authorization policies of the service provider. Those authorization policies are the basis of the enforcement of the part of the data handling policies related to access control. No mapping to concrete authorization policies is required as the data handling is expressed towards the data provider in a fine-granular way already that can be directly used for authorization at the data recipient side. In our model above, an additional mapping may be executed such that more abstract data handling policies are the basis for the communication of policies and their negotiation. The obligation part of the data handling policies is pushed to an obligation engine (data LCM component) that takes care of their enforcement. See Chapters 11 and 14 for details on the concrete model used in PRIME and its prototype and details of the resulting architecture.

9.9   Negotiation – Exchange of Data

Negotiation is the mutual request and exchange of data and other resources and relevant metadata and agreement on data handling policies for those. The purpose of a negotiation is that both parties can receive the information about the interaction partner in a way that the parties' policies are fulfilled for the interaction at hand. The exchanged information serves the purpose that the receiving party can be assured of certain properties (attributes) of the other party as required by its policies.

Negotiation is particularly interesting within settings where one party is completely unknown to the other party and the other party is possibly known to the one party and the parties not having engaged in an interaction before. This is the common case of a user interacting with a service provider over the Internet, where the latter can be a well-known entity. Mutual identity information needs to be established from scratch except for possibly some pre-existing knowledge of the parties about each other. In a negotiation, a party involved in the protocol can obtain data about the other party from the other party itself and from third parties.

In today's Internet, users can perform some kind of negotiation process that is inherent to most electronic interactions themselves: they can assess the trustworthiness of a service provider they are interacting with and they can provide attribute data about themselves to the service provider. All of this is done manually by the user with lacking support of an identity management system. The assessment of the service provider requires her to check certifications, e.g., seals the service provider claims to have been awarded as listed on

their Web page; or she may want to search the Web or specific reputation services for reputation-related information about the service provider, including reviews by customers of the service provider in natural language or aggregated ratings of the service provider. When providing her attribute data, she should beforehand scrutinize the privacy policy of the service provider herself and decide on whether it matches her preferences and thus whether to agree to it; then she has to manually enter attribute data in a Web form or use the support of a form filler. In practice, most users do not spend much of their time for assessing the service provider or for understanding the privacy policy – they simply reveal the requested data after clicking through the legal terms and privacy policy in exchange for getting a service. The negotiation protocol of our architecture is intended to fix those shortcomings and support users in their assessment of other parties before releasing data to those. This is done through a semi-automated process of mutual data exchange and data handling policy agreement between the interacting parties: the negotiation protocol. We want to note that the negotiation protocol can provide the information for a foundational mutual assessment, though, a user might want to include further information about the service provider in her final decision on whether to interact with it, including brand reputation or word of mouth.

9.9.1   Overview

A negotiation is executed between parties, e.g., a user and a service provider, in a variety of situations, all of which benefit from or require a bilateral exchange of data. Prominent examples are given next:

- a user requests a service from a service provider she has never interacted with before;
- a user accesses data, with herself being the subject, held by a service provider, excercising her right of data access as defined in [Eur95];
- a user assesses properties of a service provider's system and infers certain aspects of trustworthiness of the service provider;
- a user establishes a connection with another user (peer);
- a user requests a resource with a more restrictive authorization policy after a successful negotiation has been executed with the same service provider for a previous access of a resources;
- a service provider propagates an update of attribute data received by a user to another service provider that requires these data for executing a business process;
- a service provider releases data about one of its customers to another service provider as required by the business process.

The data being exchanged in a negotiation about the parties are of diverse nature and may, among others, comprise:

- data on the civil or legal identity of a person: first and last name, city of residence, date and place of birth, age or age range, gender, registered company name, legal form of company, CEO of company;
- other identity data about the person: personal preferences, salary or salary range, current employer;
- data on the assurance state of the party's data processing system: availability of integrity-protected boot, patch level of the operating system, availability of certain crucial security components such as a virus scanner and intrusion detection system, availability of life-cycle data management technology; a summary figure allowing a party to assess the overall system trustworthiness;
- data on certifications issued to the party: a privacy seal from an independent issuer certifying proper privacy practices being in place or a certification indicating proper business conduct in the past;
- reputation information about the party: aggregated information on the party's past behaviour in terms of business conduct or privacy; such data are usually aggregated by and obtained from third parties.

The attributes exchanged in a negotiation may be attributes with static character such as the civil identity attributes of a user, as well as attributes with dynamic character, e.g., such describing certain state information of the data processing system of a party or its current reputation status. Conceptually, both are handled in the same way, that is, are modeled through our data model and treated technically equally in a negotiation protocol and in the specification of the related policies. Technically, the implementations may be quite different as dedicated components may be responsible for handling different sorts of dynamic attribute data.

As specific kinds of attributes in PRIME's architecture, information on the state of a party can be queried by an interaction partner such that the interaction partner can assess certain aspects of trustworthiness of the party. Particularly, two components can provide and verify trust and assurance information: The *Platform Trust Management* and *Assurance* components. In the implementation of the PRIME Architecture, those components provide further functionality such as implementing independent protocols for performing a trust assessment of the party and its information processing system as well as an assessment of proper enforcement of data handling policies. See Chapters 17 and 16 for details on those components and the functionality they implement. See also the dedicated PRIME Architecture documents for details on how they precisely integrate with concrete versions of the PRIME Architecture. In the further discussion of the architecture, we completely abstract from the origin of the attribute information concerned in a negotiation as the negotiation concepts are generic and independent of the kinds of attributes being exchanged.

The concept of negotiation is closely related to the established concept of *trust negotiation* protocols. Though, we think that the term is misleading

as trust is nothing that can be negotiated. Thus, we use the term "negotiation" as this reflects better the process being executed: a multi-round protocol comprising request and release of data and agreement on data handling policies. This reflects, at least to some extent, the concept of a negotiation as commonly understood, of course in a more restrictive view. Thus, we will refrain from referring to this as trust negotiation in the discussion of PRIME's architecture.

9.9.2   Negotiation Model

We next describe our model of negotiation, particularly the message format as well as message dependencies. There are conceptual choices to be made when defining a negotiation protocol, dependent on the intended application of the protocol. The following aspects were driving our protocol design: Our protocol is particularly suitable for the interaction of a user with a service provider, with human intervention on the user side. The characteristics of such interactions are that the user is previously unknown to the service provider and the service provider may be publicly known. As a human is involved on one side, this immediately requires that the number of times and the modality of the human's interaction within a single instance of the negotiation protocol is constrained in order to not compromise usability. As another facet of our protocol, stronger focus is on protection of the user's data handling policy than the service provider's as revealing one's policy may lead to linkability of the interaction with one's other interactions already. At the start of a negotiation protocol, the user may already know some information about the service provider, e.g., its name and Web address, but not have other relevant attributes such as reputation scores, information on privacy seals awarded to it, or attributes reflecting an assessment of its platform integrity.

9.9.2.1   Message Exchanges

During a negotiation, in our model, messages m_i of the form $m_i := (q_i, s_i)$ are exchanged between the negotiation components of the parties. The message part q_i is the *request part* of the message, the part s_i is the *response part*. The request part of a message m_i is of the form $q_i = (d'_i, \rho'_i)$ with d'_i being the data request and ρ'_i the proposed data handling policy the party is willing to enforce on the data, if provided by the other party. The response part comprises a set of responses, each being a tuple (d_i, ρ_i, u) of data d_i, data handling policies ρ_i, the associated data handling policies the sender of m_i requires to be enforced by its recipient on d_i, and round index u indicating the round of the request this tuple is a response to. A message can also be a special *terminate* message to terminate the protocol before its successful completion.

The following example sequence shows the sequence of messages of a negotiation protocol with n rounds, with n even:

$$
\begin{aligned}
m_1 &= (q_1, s_1) & A &\to B \\
m_2 &= (q_2, s_2) & A &\leftarrow B \\
m_3 &= (q_3, s_3) & A &\to B \\
&\ \cdots & &\cdots \\
m_k &= (q_k, s_k) & &\cdots \\
&\ \cdots & &\cdots \\
m_{n-2} &= (q_{n-2}, s_{n-2}) & A &\leftarrow B \\
m_{n-1} &= (q_{n-1}, s_{n-1}) & A &\to B \\
m_n &= (q_n, s_n) & A &\leftarrow B
\end{aligned}
$$

Party A triggers the negotiation by requesting resources from party B through request q_1. The request can be accompanied already with a data handling policy if the accessed resources are data. Optionally, data can be already provided in the request through s_1 optimistically, in anticipation of a later request by B. Party B responds with a message m_2 requesting resources through q_2 and optionally (optimistically) providing data through s_2 or (partially) responding already to q_1. A request of a party is computed based on the requests it has received from the other party and a response is computed by responding to not-yet-answered parts of requests received from the other party.

The basic idea underlying the negotiation protocol is that each request q_k by a party that is part of m_k leads to a counter-request by the other party unless the other party can already completely fulfill the request by releasing the resources the other party had asked for. In case no data are provided proactively, that is, without having been requested, a response part s_k comprises (partial) responses to a subset of the set of previous requests $\{q_{k-1}, q_{k-3}, \ldots\}$ targeted at the party. Each new request q_k of message m_k takes into account the message exchange up to, and excluding, m_k. Note that q_1 and s_n are special in that they are related to the triggering of the negotiation and providing the requested resource. If the resources requested through q_1 are services and not data, the message s_n of the negotiation protocol is representative for providing the requested resource, although the actual resource provision may comprise a multi-round message exchange after the negotiation protocol or a service provided non-electronically. If the resources are data, s_n fulfills the request q_1 by comprising the requested data with attached data handling policies much like in any other round.

Assume an instance of a negotiation protocol is triggered in round 1 by a user requesting a service from a service provider; in round 2, the service provider requests the data it needs about the user in order to be able to provide the service; in round 3, the user requests data about the service provider it needs to know for being able to assess it; in round 4, the service provider responds with the requested data; in round 5, the user provides the data requested by the service provider in round 2. Round 6 represents the resource provision by the service provider to the user. A typical real-world protocol flow

like this would have 6 rounds, already including the initial resource request and the final round of resource provision. The human user would need to be involved into decisions in rounds 3 and 5 to influence the responses and requests on the user side. For a further assessment of the service provider by the user, the protocol can comprise an additional sequence of message pairs implementing an interactive querying of the service provider by the user, e.g., for its assurance information. Such additional steps integrate well with our negotiation model and allow for better flexibility and interactivity for a user of certain aspects of the negotiation protocol.

9.9.2.2 Termination

The negotiation may terminate with success or with failure. In the case that in round k party $\mathcal{B}$ provides to party $\mathcal{A}$ the resources requested in the initial message m_1, the negotiation terminates with success and $n = k$, that is, round k is the final round. In this case, both parties have received the resources they need with respect to the resources requested by the respective other party. More concretely, party $\mathcal{B}$ has received everything from party $\mathcal{A}$ that $\mathcal{B}$ needs in order to authorize the release of the resources requested through m_1, and $\mathcal{A}$ has received everything from $\mathcal{B}$ that it needs for releasing what $\mathcal{A}$ had requested in its requests. The final message m_n always comprises, by definition of the final message being the last message of a successful negotiation, an empty request part.

   If, in any round k of the negotiation a party is not able to release any of the data requested previously or in the current round by the other party, even if the other party would provide data requested by the party, the party sends a *terminate* message to the other party to terminate the negotiation with failure. In this case, the parties may already have released data to each other during the performed part of the protocol. The termination semantics mandates that in this case each party deletes the data so far received from the other party, with the following exceptions: a user may keep the data about a service provider for their data track and record the failed interaction; a service provider may keep data they need to retain for legal reasons. This termination semantics is inspired by the data protection legislation of the European Union; it should always be defined such that it is compliant with applicable law.

9.9.3 Policy-Driven Negotiation

For this work we focus on an instantiation of our negotiation model which is based on authorization policies protecting the resources of the involved parties and determining the negotiation messages. See Section 9.7 for an overview on the underlying authorization model we use. In this model, each party has defined their authorization policies protecting their resources such as services and data in a declarative way thereby specifying which data are required about a requesting party before the resource can be released to this party.

Furthermore, data handling policies are specified by a party on its resources specifying how the resources need to be handled by their recipients and how the party agrees to handle resources received by other parties. A negotiation starts with party $\mathcal{A}$ requesting a set of resources from party $\mathcal{B}$, expressed through a request d'_1. A structure of authorization policies of party $\mathcal{B}$ may apply to each of these resources, their composition leading to a data request expressed through a formula d'_2 that $\mathcal{A}$ or other parties need to respond to in order that $\mathcal{A}$ can access the resources specified through d'_1. Such a request d'_2 can be considered a request of a set of resources that $\mathcal{B}$ requests from $\mathcal{A}$ or obtains from other parties. At $\mathcal{A}$, a structure of authorization policies may apply to each of the resources requested by $\mathcal{A}$, analogous to the situation as discussed at party $\mathcal{B}$. After some rounds of mutual requests, in the best case already before the first request, a party may be able to release some of its resources that the other party has requested if the party's authorization policies permit it to do so. These released resources may "unlock" resources of the other party as now some of the other party's authorization policies may become fulfilled. The explained mechanics of a negotiation show the strong dependence of our negotiation protocol on our authorization model.

Negotiations optionally involve, besides authorization and data handling policies, another kind of policies that we denote *negotiation policies.* Those policies have the following functions: defining priorities over data in terms of its release, e.g., to prefer to use the electronic driver's license to the national id card or the passport to release or make statements about the attributes first-name, lastname, or birthdate; specifying which data may be released proactively under which circumstances, e.g., without being requested, an example being the automatic release of a subscription to an on-line newspaper when requesting an access-protected resource from this party; specifying cases in which the human need not be involved in a negotiation, e.g., for repeated logins to the same account. We leave a concrete specification of the negotiation policies open in the architecture and only explain at which points in the architecture they are of relevance. A concrete definition of a language for such policies is left to future work that integrates with our architecture.

9.9.4   A Round of Negotiation

We discuss next in detail the processing of a single round of negotiation of a party while giving attention to optional steps at the user side for interaction with the human user. Except for those user interactions, the process is the same for any party, e.g., users or service providers. Service providers in practice do not require any manual intervention during the execution of a negotiation protocol. Let the first party be denoted $\mathcal{A}$ and the other party be denoted $\mathcal{B}$. Let the round we consider be the k-th round of the negotiation protocol explained from the perspective of $\mathcal{A}$.

The processing requires a sequence structure Q maintained throughout an instance of the protocol for keeping records each comprising a received request

and a computed response to this request that cannot be fulfilled at the time of processing. The processing furthermore requires a data structure Q^* for a party that captures the requests it sends, that is, the requested data and proposed data handling policies on those.

At the beginning of a new round, the party has all data and data handling policies received in previous rounds in the dynamic subject profile for this protocol instance. Furthermore, it has a record of all requests and data sent to the other party and requests received from the other party within the negotiation. For notational simplicity we leave implicit the variable instantiations that need to be considered when checking whether a formula fulfills a request and leave it up to an implementation to keep track of it. We next present the details of the processing steps a party performs in the k-th round of a negotiation protocol instance.

Receiving the Message

At the start of round k, the party receives the input message m_k of the form (q_k, s_k). The element q_k comprises a request for resources d'_k and associated proposed data handling policies ρ'_k. The element s_k comprises a set of tuples $\{(d_{k,j}, \rho_{k,j}, u_{k,j})\}_{j=1..w_k}$.

Processing Received Data and Data Handling Policies

The party processes each tuple $(d_{k,j}, \rho_{k,j}, u_{k,j})$ of the response element of the message as follows. In case that the provided data and data handling policies $(d_{k,j}, \rho_{k,j})$ have not been sent by the other party in response to an earlier request by the party, but proactively, the following processing is performed: the party checks whether $\rho_{k,j} \preceq \rho$, that is, the data handling policies associated with the received data subsume the party's data handling policies ρ for the concerned data (types) of $d_{k,j}$ as held in the party's policy repository.[21]

In case the provided data and policies $(d_{k,j}, \rho_{k,j})$ are sent in response to a previous request, the party retrieves from Q^*, using the round index $u_{k,j}$ for the request provided with the data, the data handling policies $\rho'_{u_{k,j}}$ for the data $d_{k,j}$ proposed previously in round $u_{k,j}$ by the party together with the data request $d'_{u_{k,j}}$ to which $d_{k,j}$ is a response. The response $(d_{k,j}, \rho_{k,j})$ of the other party may fulfill the request only partially in this round. Let $\rho''_{u_{k,j}}$ that are a part of $\rho'_{u_{k,j}}$ be the data handling policies corresponding to the data in the partial response $d_{k,j}$. The party checks whether $\rho_{k,j} \preceq \rho''_{u_{k,j}}$, meaning that $\rho_{k,j}$ are compliant with the originally-proposed ones $\rho''_{u_{k,j}}$. If this holds, the data release by the other party fulfills the original request of the party from the perspective of the data handling policies. If not, the data handling policies are not acceptable to the party and it terminates the protocol. The

---

[21] In other words, the party checks whether the proposed data handling policies for the provided data match its own policies that it would propose when requesting those data from the other party.

policies $\rho_{k,j}$ are the policies that need to be applied by the party on data $d_{k,j}$ if the protocol succeeds and the data are kept.

If the formula in $d_{k,j}$ comprises, optionally, more data than requested in $d'_{u_{k,j}}$, such as additional attributes, the additional items may be covered by additional data handling policies in the party's policy repository for which no proposal has been made by the party in its original request. For those, the party needs to decide whether it wants to accept the data handling policies by checking with its own policies it would propose in a request for these data in the policy repository, like in the case above. In case it cannot accept them, it terminates the protocol with failure. Thus, the case of providing additional data combines the two described cases of a response without a request and a response to a request.

The party adds the formula $d_{k,j}$ to the dynamic subject profile associated with the other party and keeps track of the associated $\rho_{k,j}$. For the remaining rounds of the negotiation, the data are—due to being in the dynamic subject profile—considered for every evaluation of an authorization request by the party.[22]

Processing of Partial Requests in Q

The above step of adding data about (from) the other party to the dynamic subject profile may have "unlocked" resources of the party for release that have been requested earlier by the other party. In this step, the party checks whether it can release data in response to elements $((d', \rho'), (d, \rho))$ contained in the sequence Q of requests created in one of the previous rounds. In such a tuple, (d', ρ') is a previous request by the other party and (d, ρ) the part of the response to the request that still needs to be fulfilled by the party. Parts of the request may have already been fulfilled in previous rounds. For the request in the tuple, a decision has already been made in the round when the request was received on how the request is to be fulfilled, though no authorization was available so far for releasing the resource. We perform the processing as explained next for each such tuple in Q, starting from the topmost index. Let (d, ρ) be the current element being processed.

Checking whether (parts of) d can be released is done by performing authorization decisions on the resources of d and, in case of only *grant* responses from the authorization engine, d can be fully released under ρ in the next message sent to the other party. If the authorization queries all return *grant*, let $d_r = d$ and $\rho_r = \rho$. In case of at least one data request being returned as responses to the authorization queries, split d by creating a formula d_r such that $d_r \wedge d_r^* = d$ and split ρ such that $\rho_r \cup \rho_r^* = \rho$ and ρ_r are the data handling policies to be applied to d_r and ρ_r^* the data handling policies for d_r^*. Those

---

[22] Note that at this point data are not yet added to the subject profile related to the other party as it is unclear whether the negotiation will be successful. This is only done in the post-processing once the negotiation has been concluded with success.

elements are computed such that d_r comprises as many as possible of the resources for which the authorization query returned a *grant* response. Add (d_r, ρ_r, u) to the response set s_{k+1}. u is the index of the round of the request being answered. If $d_r \preceq d'$ holds, that is, d_r is a subformula of d', remove the current element from sequence Q as in this case, the complete request can be successfully responded to and does not need any further processing. If the condition does not hold, that is, the request can only partially be responded to, remove the element of Q being processed and add the following element in its place: $(d', \rho'), (d_r^*, \rho_r^*)$. This specifies that (d_r^*, ρ_r^*) still needs to be released in a future round of the protocol in order to completely fulfill the request (d', ρ'). We note that splitting a data statement as described above must adhere to the data semantics of our data model and limitations can apply, e.g., splitting of a disjunctive statement that must be proven as a whole is not possible.

This approach of allowing the other party to answer a data request in part and then the party revealing as much data for a previous request as it can, leads to a protocol that has less potential of getting deadlocked due to no party being able to respond to the other party's requests any more for too restrictive authorization policies.

Processing of a New Request

This step processes a new request q_k of the current round and computes a decision on how the request can be responded to. The party needs to compute how and whether it can fulfill the request $q_k = (d'_k, \rho'_k)$, that is, find a data formula d'^*_k as well as data handling policies ρ'^*_k fulfilling the request, and construct a data request to send as a new request q_{k+1} to the other party based on authorization policies applicable to the computed response d'^*_k. This new request needs to be fulfilled by the other party before the party can fulfill q_k. We note that the choice of d'^*_k and ρ'^*_k are interdependent as both the data request and the data handling policies must match the request. Thus, it may be useful to allow for reverting to choose a different data formula in case no agreement on the data handling policies can be found for a particular choice of formula.

The request d'_k is a formula in our data model which precisely specifies the data that the party needs to provide. The party performs a matching operation with its stored data, that is, its identifier relationships, identity relationships and profile data, and data handling policies and finds a tuple of all formulae $d^*_{k,i}$ that can fulfill the request as well as the party's data handling policies $\rho_{d^*_{k,i}}$ retrieved from the policy repository for each formula such that $\rho_{d^*_{k,i}} \preceq \rho'_k$ holds. Thus, all elements of the tuple fulfill the request q_k from both a perspective of data and data handling policies. Based on the tuple of formulae $d^*_{k,i}$ and the associated data handling policies $\rho_{d^*_{k,i}}$, a decision on a concrete element $d^*_{k,y}$ of the tuple and its associated data handling policies $\rho_{d^*_{k,y}}$ needs to be taken next and a negotiation of the data handling policies needs to be performed.

The decision for a concrete element $d^*_{k,y}$ of the potential candidates of statements to be released and the negotiation of associated data handling policies based on $\rho_{d^*_{k,y}}$ can be based on negotiation policies of the party, user input, and possibly other decision mechanisms. Concretely, the formula $d^*_{k,y}$ is chosen that is best suitable regarding the party's negotiation policy, user input and other mechanisms of decision. The associated data handling policies $\rho^*_{d^*_{k,y}}$ are the result of an instantiation of $\rho_{d^*_{k,y}}$ such that $\rho^*_{d^*_{k,y}} \preceq \rho_{d^*_{k,y}}$ and $\rho^*_{d^*_{k,y}} \preceq \rho'$. This step constitutes a negotiation of the data handling policies based on a proposal of the other party, the data handling policies of the party, and possibly human user input and other decision mechanisms. We chose to not include interaction in this policy negotiation in order to not expose additional preferences of a user to a service provider and thus risk to harm privacy and also to reduce protocol complexity.

At this point, the party has a preliminary decision on the data and policy response to q_k. The formula $d^*_{k,y}$ constitutes at least one resource of the party. The party needs to determine the authorization decision on the release of the formula or its parts to the other party. The party invokes the authorization engine for all resources expressed in $d^*_{k,y}$; see further below for how this can be done in our general model of policy associations with resources. Each invocation of the authorization engine on a single resource can result in a *grant* or *deny* response or a data request. In case of only *grant* responses from the authorization component, the data $d^*_{k,y}$ related to the request q_k can be released to the other party under data handling policies $\rho^*_{d^*_{k,y}}$ and the data request d'^*_k targeted at the other party is the empty request. In case of one or more *deny* responses, the party is unable to fulfill the request in its entirety and terminates the protocol or rolls back the processing and reverts to a different choice for $d^*_{k,y}$ and $\rho^*_{d^*_{k,y}}$ in the previous sub-steps of the computation. In case of at least one data request resulting from the authorization requests and no *deny* response, the party splits the formula $d^*_{k,y}$ into the parts $d^*_{k,y,t}$ and $d^*_{k,y,f}$ such that $d^*_{k,y,t} \wedge d^*_{k,y,f} = d^*_{k,y}$ and such that the data handling policies can be split accordingly. It splits the data handling policies $\rho^*_{d^*_{k,y}}$ into $\rho^*_{d^*_{k,y,t}}$ and $\rho^*_{d^*_{k,y,f}}$ such that each policy structure comprises the data handling policies for one of the parts of the data. Thereby, $d^*_{k,y,t}$ are the data that can be released, following the authorization queries, at this point in the negotiation and $\rho^*_{d^*_{k,y,t}}$ are the associated data handling policies. Analogously, the formula $d^*_{k,y,f}$ is the one for which authorization queries have resulted in requests for data. A combined data request d'^*_k is created by composition of those data requests. A proposal for the data handling policies ρ'^*_k for a response to d'^*_k is created by retrieving the policies for all resources of d'^*_k from the policy repository.

The computed intermediate results together with the request are added to Q as the tuple $((d'_k, \rho'_k), (d^*_{k,y,f}, \rho^*_{d^*_{k,y,f}}))$ for use in a later round when (parts of) the data $d^*_{k,y,f}$ can be released. The data $d^*_{k,y,t}$ can be released already in

this round, under the data handling policies $\rho^*_{d^*_{k,y,t}}$, this is done in the next processing step.

Create and Send Message

A new request q_{k+1} targeted at the other party is created: It is constructed from d'^*_k, and ρ'^*_k as the corresponding data handling policies. The tuple $(d^*_{k,y,t}, \rho^*_{d^*_{k,y,t}}, k)$ is added to the response part of the message. The message $m_{k+1} = (q_{k+1}, s_{k+1})$ is sent.

Note that for reasons of associating the different parts of the corresponding requests at the other party and thus simplifying the processing, the responses are not combined into a single formula which would be feasible from the perspective of our data model. From a pure authorization perspective at the recipient side of the data, a single formula would be sufficient, the processing would be more complicated, though, as the relation to its corresponding requests would be missing.

9.9.4.1 Specific Aspects of the Protocol

Post-negotiation Processing

Once an instance of a negotiation protocol has successfully terminated, the following operations need to be performed on the data obtained in the negotiation: Firstly, the data received in the negotiation that rest in the dynamic subject profile need to be made persistent by storing them in the profile record associated with the other party. The data handling policies are stored in the policy repository and refer to the data they apply to as outlined in Section 9.8. Secondly, the enforcement of the data handling policies needs to be initiated: The parts of the data handling policies that define authorization requirements at the party are mapped into according authorization policies; either those already exist as policies on data categories, or they need to be created on the instance data. The parts of the data handling policies that define obligations must be pushed into the data life-cycle management component that takes care of their enforcement. The data handling policies for the received data are used whenever data are to be disclosed to third parties by the party to agree on compliant data handling policies with a third party, that is, it is ensured that the policies that are agreed with a third party are at least as strict as the data handling policies agreed in this instance of the negotiation protocol. For this, the party will use the data handling policies as a proposal to the third party much like the other party that has released the data to the party has done.

If the negotiation terminates without success, a service provider may store only user data that it needs to store for legal reasons, the default it to not store data. A user may store all the data about service providers for data tracking purposes, following European data protection legislation. When users store

data about other users, provisions of the European data protection legislation may be applicable as well as they then act as a data controller from a legal perspective.

Computation of a Response: Usability

A step of substantial complexity in the execution of a round of the protocol is the creation of the data formula $d^*_{k,y}$, the response to a request. On the user side, this is a process that first determines all possible formulae how the request d'_k can be fulfilled with locally-held data—mainly identifier and identity relationships—and, based on the possible formulae, involves the human user in making choices based on her preferences. Finding the set of fulfilling formulae is explained in Section 9.3 on the data model 9.3 and is based on operations on formulae within our logic. The associated data handling policies are retrieved from the policy repository. The difficulty comes in when the possible choices of formulae and policies need to be presented to the human user and she needs to interact with the system to make her choice and give her informed consent. The general form of the protocol as described above may lead, depending on the authorization and negotiation policies of the parties, to multiple points where data releases are performed by a user within a single instance of a negotiation protocol. This is a particular challenge from a user interface perspective as multiple user interactions in a single negotiation instance are not per se intuitive for a user. We think that an appropriate way of designing the user interface is to cumulate the information of the multiple rounds into the user interface and get the user's consent for the individual parts. The protocol guarantees that the information known about the other party, which is important for informed consent as it determines the recipient of the data, is growing monotonically which simplifies its display. As a specific feature for allowing a user to better assess a service provider, an interactive querying interface may be thought of. Proposals for user interface concepts for data release have been made in Chapter 20 of this book. The proposal presented in [CSSZ06] shows another interesting interface that integrates with a Web browser and allows for simple identity selection through user actions. Though, those proposals for the interface do neither consider multi-round decisions and consent of a user nor formulae of the complexity as ours and thus are only applicable to a much simplified (restricted)—but also practical—variant of the negotiation protocol. We want to note that already the user interfaces for these simplified protocol variants raise many issues in terms of how to best build such an interface. Those issues likely transfer to the more complex protocol as they are of conceptual nature, and additional issues will arise from the greater complexity of the proposed protocol and language.

Determining the Resources Comprising a Formula

The processing in a negotiation requires, as explained in the protocol, that a party computes an authorization decision for a formula d comprising at least

one resource. As explained in Section 9.7, authorization policies can be associated to data at different granularities: single predicates, (sub-)formulae, or classes of those. This means that for a composite resource, different policies can apply to a single resource on those different granularities. The processing for a formula ϕ is done, explained non formally, as follows: The formula is represented as a tree, with the logical connectives forming its inner nodes and predicates the leaves. The tree is traversed starting from its root in a traversal as follows: the current node is processed, then its subtrees are processed recursively. For a node, we check whether there exist authorization policies that apply to the subtree rooted in the node as a whole or a part of it. If this is the case, this part is considered as one resource with the applicable policies. If the node is an $\vee$-connective, the subtrees of the node are not further processed as the tree is an atom from the perspective of policies, if it is an $\wedge$-connective, the algorithm is continued recursively on the subtree. As final result, the algorithm outputs the not necessarily non-overlapping resources of the tree for which authorization policies are defined. When releasing the formula to the other party in an interaction, all the found authorization policies need to be considered for d due to the general mechanism of associating policies with resources.

The purpose of this seemingly complex processing of a resource is to implement the definition of authorization policies on different granularities of resources. This allows, for example, to define a more restrictive policy for releasing certain $\wedge$-combinations of attributes in addition to the policies applicable to the attributes when released alone. Note that subtrees having a disjunction at their root that is to be proven as an atom are not further processed as it would not reflect the correct meaning of authorization policies to apply the policies to the disjunctive parts of the tree and create data requests based on them.

9.9.4.2 Negotiation Messages vs. Data Exchange

We note that in the above negotiation protocol, data exchanges are modeled by simple message exchanges. In a practical execution of a negotiation, though, cryptographic or other protocols for the exchange of data are executed in addition to the message exchanges whenever a party provides a data response for releasing the data of s_k. Such protocols are abstracted in the discussion of the negotiation component for a clean separation of the decision on identity release and the mechanics of release without a loss of substance in the presentation. The negotiation component assumes that all data received is appropriately endorsed as specified in the data through the certification metadata, that is, all formulae are shown to hold. See Section 9.6 for details on data exchange and Section 9.3 for certification metadata.

Considering the above additional aspects of negotiation should give the reader a more complete account on the negotiation protocol and particularly its integration into our architecture.

9.10   Conclusions

The architecture we have defined in PRIME provides a comprehensive treatment of user-centric identity management, with foci on data minimization through the use of private certificate systems and the related integration of required technologies. The key goal of the work has been to define a practical system with the potential of technically realizing the concept of informational self determination and strengthening the trust model compared to today's systems. In our work on such a system, we were to resolve different systems-related and integration aspects that arise when bringing the used technologies together.

The work on the architecture has built on available results in different fields and improved them with respect to aspects that are crucial for achieving our privacy goals. This has lead for example to the creation of our powerful data model, the generic interface of the data exchange component, the extensions to the authorization system we build on, and also our negotiation protocol based on attribute-based authorization. Those underlying technologies have been integrated into a system in order to provide user-centric identity management functionality to parties.

9.10.1   Key Contributions

We next summarize our main contributions on the architecture we have made with the definition of and work on the PRIME Architecture.

Our *data model* allows for formally representing attribute statements between parties in a data-minimizing way throughout the architecture. Thus, it forms the backbone of our architecture: The data model is the common formalism which is used for linking together different components as well as for different parties to communicate with each other. This model improves on available techniques for representing attribute data in terms of expressiveness of data minimizing statements and the capability for automated derivations. Also the aspect of delegation is captured in our data model such that authorization policies can be expressed for delegation use cases and those can be handled by the complete system.

We have specified *generic interfaces* for data release protocols and prototyped them on the example of the Identity Mixer private certificate system. Those interfaces integrate with the data model and thus allow for private certificate systems to be integrated with our architecture. This has been one of the main objectives for the PRIME Architecture.

Based on a *privacy-enhanced authorization* system, we have specified extensions to this system that allow for stronger data minimization than the original system, thus being an integral part of the architecture. Particularly, the expressivity of our data model has been made available to the authorization system for expressing properties of requesters. This extended authorization system is the basis of our negotiation protocol for mutual exchange of data between parties in an interaction. This negotiation protocol is practical yet powerful by the trade-off between real-world practicability and system complexity.

Another contribution is the specification of the representation and use of the different kinds of *data held by parties* using the unified data model. This is an aspect that is partially related to implementation, but also key to the integration of technologies, e.g., the use of identity relationships for fulfilling data requests.

The architecture also integrates with life-cycle data management solutions that have been built within the PRIME project, as well as trust and assurance assessment solutions, with the main integration point being the data model. Those components have been integrated into our prototype implementations that have been done on fragments of the architecture functionality in order to reduce their complexity. Details on this can be found in previous architecture documents and related papers authored during the PRIME project.

A major contribution of the architecture is to bring selected privacy-enhancing technologies together and the *specification of the interplay* of those. This has encompassed an effort spanning multiple disciplines, such as cryptography, attribute-based authorization models and languages, negotiation, and logic in order to define the "glue" that orchestrates the technologies in a way that the intended privacy-enhancing functionality is provided to parties as well as to extend the technologies accordingly to match our overall architectural model. Through the integration of private certificate systems, a major aspect of our work, we were able to reduce the trust requirements in other parties as far as data exchange is concerned.

9.10.2 Experience

During the process of developing the PRIME Architecture, the team involved in the work has gained *substantial experience* in the area of integrating multiple privacy-enhancing technologies with the goal of orchestrating them such that our goals of privacy protection are reached. We have experienced a rather high complexity on the technological side in our architectural efforts, but could master this complexity and have succeeded in the integration efforts with the result being a comprehensive architecture for privacy-enhancing identity management that follows the user-centric paradigm. Overall, the architecture effort was touching on multiple quite different technological disciplines that needed to be understood in order to define the architecture.

On the requirements side, we have experienced a very *diverse landscape of requirements* in different domains and at different levels of abstraction for a system like PRIME. In the early phases of the project it was a challenge to sort out the key requirements that should drive our technology development. Over time, it became quite clear what the key requirements are and our work was driven to a large extent by those. Particularly difficult were requirements at the social and economic level as they can mostly not be directly accounted for by technology, but rather require a concerted approach of various disciplines.

Future Work

We mention some areas of future work that we think are of importance to be addressed in the area of architecture of user-centric identity management in addition to the treatment they have received in PRIME's architecture efforts.

In terms of *basic research*, there is potential for all the involved technologies to be improved or extended, e.g., to provide for even stronger privacy features or improve their efficiency. Such extensions may comprise also changes in the architecture, e.g., additional message flows or changes to the data model to accommodate the new features. The current design of the architecture is expected to anticipate such future changes to a certain extent.

Attribute-based delegation is supported by the current architecture, though it can be realized by exploiting features that have not specifically been designed for delegation. For example, expressing delegation relations in authorization policies is possible, but not as easy as it should be. Dedicated language features and support in our model would improve on this situation and are thus an area of future research.

As mentioned already in the text, *usability* is an important area of ongoing future work because current work is still unsure about the right interaction paradigms to use for various aspects of user-centric identity management. Future work will be needed for identity selection using the generic approach we discuss and to explore the basic paradigms such as the identity card metaphor, on-the-fly policy specification, and assessment of information about the other party in an interaction, to name some key areas. Usability is probably one of the most relevant areas of research required for a successful future deployment of user-centric identity management technology.

A major area that needs to be tackled in the mid term are aspects related to *deployment* of user-centric identity management technology. This particularly comprises standardization of used technologies and defining sufficiently simple fragments of our architecture that can be used in first deployments. Aiming at deployment of the the full feature set of our architecture at once probably raises the issue of being too complex, and suitable, but much simpler, fragments can already provide substantial improvement of privacy protection for users compared with today's situation. As particular drivers for real-world deployments, business models for privacy may be of particular relevance.

To conclude the architecture chapter of this book, we want to claim that our work on the PRIME Architecture for user-centric identity management has been a sustantial step forward by conceptualizing this area and by providing a description of a practical system for protecting the privacy of the citizen. Particularly, our architecture is a step towards bringing the European society closer to *informational self-determination* of the citizens for electronic interactions and supporting them in exercising their rights granted to them by the European Data Protection Directive [Eur95]. Features of our architecture are to a large extent based on the European legal data protection requirements and its implications – one of the strictest data protection legislations in the world. Through the use of private certificate technology, particularly the Identity Mixer system, for attribute exchange, we have strengthened the underlying trust model as much as possible with latest available technology. As the basic technology foundation has been defined now through the PRIME Architecture, what is most needed for a successful future deployment is resolving issues in the legal, business model, usability, and social areas to take the next steps in a successful deployment of PRIME's architecture and technology.

10

Pseudonyms and Private Credentials

Jan Camenisch[1], Markulf Kohlweiss[2], and Dieter Sommer[1]

[1] IBM Research
[2] KU Leuven

10.1 Introduction

In this section we are concerned with cryptographic means for protecting the privacy of users in electronic transactions. That is, our goal is to enable the user to conduct transactions while revealing as little information as possible. Of course, in most transactions, a user needs to reveal some information. Hence our goal will be that the user need not reveal any information in addition to what is necessary to conduct the transaction. Let us make an example to illustrate this point. Assume the user wants to rent a car and that in the process she needs to produce a driver's license. If she would do so today, she would just show her (paper) license and the car-rental agency would inspect it and thereby learn her name, address, etc., while it would be sufficient if the agency would be able to see only the user's picture, to verify that the license was indeed issued to the individual who intends to rent the car, and possibly the expiration date, to verify that the license is still valid. Now, we would like to achieve the same for digital certificates. If we would try to do this with conventional certificates or federated identity management tokens, we see that we would either need to enable the user to selectively reveal attributes of certificates (while hiding others) or require the user to get a certificate that includes only the attributes required for the transactions.

The latter has the disadvantage that the user is required to get a certificate for each transaction and thus involving the issuer of the certificate in each of these transactions. In this way the issuer who operates a highly critical security service becomes a single point of failure both for availability and security but also for the users' privacy. The issuer would learn a

large amount of information about the user as he is involved in all the user's transactions with a variety of different service providers and in different usage contexts. Furthermore, because of the binary representation of the certificates, the issuer and a verifier (i.e., the car rental agency) can link the respective transactions (and thus learning more information than necessary). This also holds for the XML security tokens that replace certificates in todays federated identity management systems. They can be seen as attribute certificates that are freshly created on the fly for each transaction. Thus, the former solution (where certificates are used multiple times) seems to be a better alternative. However, if one employs traditional certificates, the user cannot selectively reveal attributes asserted about her in the certificates. Furthermore, to prevent the linking of the issuing and verification transactions of certificates, the user would need to reveal the attributes in a fashion that does not reveal any other information about the certificate itself (such as its binary representation).

Theoretically, the task of unlinking issuing and verification could be accomplished by so-called zero-knowledge proofs that allow the user to prove to the verifier that she possesses a certificate containing the necessary attributes (without revealing their binary representation). However, for traditional certificates and tokens, this would not be practical as we would need to employ general-purpose proofs that are by far too inefficient. Luckily, there exist special signature schemes that can be used for issuing certificates such that we can employ particular zero-knowledge proofs that are efficient. These allow one to implement so-called private (or also anonymous) credentials systems.

10.2   The Idemix Private Credential System

Idemix, which stands for "Identity Mixer", is a strong, yet privacy-friendly authentication system based on the private credential system proposed by Camenisch and Lysyanskaya. In Idemix, each client is issued a private certificate that allows her to authenticate herself to service providers. Now, the client has a choice. She may authenticate herself by revealing all information in her certificate to the service provider, as is traditionally done, or she may choose to reveal only the necessary information to the service provider (e.g., reveal her age without revealing her name.) The client can authenticate to many service providers using only one small certificate. Because of this flexibility of Idemix works for all authentication needs, be it in a government setting or in a corporate infrastructure.

10.2.1   Basic Principles of Strong Authentication

Authentication is a method for a client to convince a verifier that she satisfies a well-defined condition. There are many forms of authentication in use today. Many traditional mechanisms only offer weak authentication such as when a client shows her employee badge to a security guard or checks her email using

a password. These mechanisms are considered weak because they can easily be circumvented: a badge can be forged or a password can be guessed. A better solution is to use the *strong authentication* offered by cryptographic authentication and identification schemes such as Idemix. In Idemix, private certificates are issued to clients to properties of the client (e.g., name, birthdate, nationality, or health records). A client maintains a portfolio of private certificates locally.

The client uses certificates from her portfolio to strongly authenticate herself to a service provider while only releasing the required information in the authentication process. That is, a client is not required to convey all information in her certificate in order to perform the authentication, rather the client can securely release only selected portions of her certificate. This new paradigm of property-based authentication enables a new world of access control policies, as clients can now authenticate certain properties about themselves without forfeiting their privacy (e.g., a client can prove she is a Swiss citizen under twenty five without revealing her name.) For each new authentication to a service provider, a client uses one or more certificates from her portfolio to create a new authentication token. This authentication token is transmitted to and verified by the service provider. Based on the result of the check and whether the authentication fulfills the access control policy of the service provider, the client is granted or denied access. We note that the private certificates themselves are never sent to any party.

10.2.2   Balancing Anonymity and Accountability

Idemix offers very strong anonymity guarantees to its clients. A Danish teenage website can now enforce a policy that only Danes under twenty may post without requiring these teenagers to give their names. And yet, what happens when a particular client starts to abuse this anonymity? Any valid authentication mechanism must have a method for dealing with abuse, and the Idemix system is carefully designed to handle potential abuses of anonymity. First, consider the scenario where a bomb threat is posted to the Danish teenage website. The police may have a compelling interest in learning the name of the client behind this post. Yet, suppose the teenager did not include their name in the authentication token and thus even the website administrators do not know it. To this end, the Idemix system includes a method to revoke the anonymity of any authentication token, and discover the client's identity. Revocation of anonymity requires knowledge of a secret cryptographic key, which can be owned by an authority (e.g., the police) or shared between two or more authorities (e.g., the police and a judge must cooperate to revoke a client's anonymity). Second, consider the scenario where a few teenagers are dominating the conversation, and thus the website administrators want to limit each person to five posts per day. If clients post anonymously, how can this policy be enforced? To address such requirements, the Idemix system allows the website administrators to announce a policy of,

say, five posts per day and then clients' authentication tokens, to properly verify, must reflect this policy. Indeed, in Idemix, a client who posts five or fewer times will remain anonymous, but if the client attempts to post a sixth time, this fact will be quickly detected and the client can be identified and appropriate measures can be taken. This mechanism also discourages teenagers from sharing or selling their access credentials, as the limitation to five posts per day will hold over the whole group of people that use the same access credential. These are two techniques by which Idemix can balance anonymity and accountability. Indeed, the Idemix system allows for a balance far beyond what is possible in other authentication systems.

In the next section we describe the concepts implemented by Idemix. In the following section we then describe how applications can be build on top of Idemix.

10.3   The Idemix System

The Idemix library (`prime.inf.tu-dresden.de/idemix`) is an implementation of an anonymous digital credential infrastructure which can help to achieve the functionality described above. It provides the bare certification functionality and does not yet provide application-specific functionality, such as mechanisms for balancing anonymity. Users can obtain private certificates, i.e., credentials, and can create proofs about them. A proof is done with respect to a specific pseudonym, but can cover multiple certificates (each potentially obtained by the user under a different pseudonym). A proof corresponds to an assertion that is expressed in XML and which can be modularly composed. In addition to statements about attributes, the assertion can involve binary cryptographic objects such as commitments and encryptions (cf. §10.3.2).

A (digital) certificate consists of *data items* and a digital signature by a *(certificate) issuer* on the *data items.* By signing the user's data items the issuer certifies for instance the user's authorization to perform some given task and that it has verified the validity of (some of) the user's data items. To demonstrate its authorization and the validity of the data items, the user can for instance show the certificate to a *verifier* who checks the certificate's validity by verifying the correctness of its signature. The verifier will accept the claims associated with a certificate as far as he trusts the issuer w.r.t. these claims.

In the following we describe desirable properties of (non-traditional) certificates that allow the user to control what data items are disclosed to the issuer and verifier of certificates respectively.

10.3.1   Required Properties When Showing a Certificate

By showing a certificate we mean the process whereby a user tries using a certificate she possesses to convince a verifier of the contents of the certificate.

We stress that during this process the user does not necessarily send the actual certificate to the verifier.

We require a process that allows the user to show a certificate such that the following properties are met:

Multi-show unlinkability: Conventional (public-key) certificates are represented (encoded) by unique strings. Thus, when the user would just send the certificate obtained from the issuer to the verifier, the issuer and the verifier can link the transactions. Furthermore, multiple showings of the same certificate to the same or different verifiers are linkable. We would like to emphasize that linkability is an inherent property of traditional certificates, which is independent of the data items contained in such a certificate. In particular, even transactions performed with so-called pseudonymous certificates, i.e., certificates that do not contain personally identifiable data items, are linkable. Linkability is known to be a serious threat to the privacy of individuals. We require that the showing of a certificate cannot be linked to the issuing of the certificate as well as to other showings of the same certificate, unless of course the data items being disclosed allow for such linking.

Selective show of data items: Given a certificate, we require that the user in each showing of the certificate can select which data items she wants to disclose (and which data items she does not want to disclose) to the verifier. For numerical data items, we require that it be possible to show that a data item lies in some interval without revealing the exact value of the data item. As an example, consider a driver's license certificate consisting of the user's name, address, and date of birth. When stopped on the road at a police checkpoint, the user shows that the certificate is valid, i.e., that she is authorized to drive, without disclosing her name, address, and date of birth. Using the same certificate, in a supermarket when purchasing alcoholic drinks, the user shows the certificate such that she only discloses that she is not underage.

Conditional showing of data items: We require that the user be able to conditionally disclose certified data, when showing a certificate. More precisely, let us assume that there is a third party, and that prior to certificate showing the user picks the data items she wishes to show conditionally to the issuer; also the user and the verifier agree on the conditions under which the verifier may learn the selected data items. In a conditional showing, the user discloses to the verifier information (on the conditionally shown data elements) such that the verifier cannot recover the conditionally shown data items from the information. Yet, the verifier can be assured that, when asked to do so, the third party is able to recover the data items.

Hence, if the third party recovers the data items only if the mentioned condition is fulfilled (where we assume that it knows the condition), then the above mechanism implements showing of (certified) data under the agreed condition.

As an example, consider a user accessing a university library's reading room with valuable books and the third party being the university administration. The user's identity, e.g., contained in her student identity certificate, will be disclosed to the librarian only under the condition that books are stolen from the reading room. To find out the user's identity, the librarian will need to involve the university administration.

Proving relations between data items: When showing multiple certificates by different issuers, the user should be able to demonstrate that data items in the certificates are related without disclosing the data items. For instance, when showing her driver's license certificate and her credit card certificate to a car rental company, the user should not need to disclose her name contained in the certificates, but only to demonstrate that both certificates are issued to a person with the same name.

Desirable Properties of Certificate Issuing

We now describe the properties we require of the process where the user gets issued a certificate by an issuer. Let $\{m_1, \ldots, m_l\}$ denote a set of data items and H a subset of these data items. It should be possible for the user to obtain a certificate on $\{m_1, \ldots, m_l\}$ such that the issuer does not learn any information on the data items of H, while it learns the other data items, i.e., $\{m_1, \ldots, m_l\} \setminus H$. We refer to such an issuing as *blind certification*.

Obviously, the data items in H are chosen by the user, however the other data items could be chosen by the issuer or by the user. For the data items that remain hidden from the issuers, we require that the user is able to assert that some of them were previously certified by another issuer. An example where this property is useful is e-cash with online double spending tests. Here the user chooses a random and unique number that is certified by the bank (issuer) such that the bank does not learn the number (cf. §10.4.3).

10.3.2   Cryptographic Primitives

In this section we illustrate how a framework of encryptions, commitments, signatures, and zero-knowledge proofs can be used to implement certificates having properties as described above. The presentation is (quite) informal and intended to be accessible for non-specialists in cryptography. We first introduce the abstract properties of zero-knowledge proofs of knowledge and then discuss encryptions, commitments, and signature schemes. In the latter we are particularly interested in schemes that allow for efficient zero-knowlege proofs.

By $\omega = A(\alpha)$ we denote that ω is output by the (probabilistic polynomial-time) algorithm A on input α.

Zero-knowledge Proof System

We consider *zero-knowledge proofs of knowledge*. Let W denote an arbitrary boolean predicate, i.e., a function that on input some value either outputs

1 (true) or 0 (false). A proof of knowledge is a two-party protocol between a prover and a verifier, where the common input is a predicate W, and the prover's input is a value w for which W is true, i.e., $1 = W(w)$. At the end of the protocol the verifier either outputs 1 (accept) or 0 (reject). The protocol has the property that if the verifier accepts, then it can be assured that the prover knows a secret value w' such that $W(w') = 1$. The protocol is zero-knowledge if the verifier does not learn any (computational) information about the prover's input w. We denote such a zero-knowledge proof of knowledge by $\mathrm{PK}\{(w) : W(w) = 1\}$. Often we use proofs of knowledge where W is a composite predicate in multiple variables. Our notational convention is that the elements in parentheses denote secret values. Through the proof of knowledge the prover convinces the verifier that he knows such values and that they satisfy the predicate. These values are (in general) not known to the verifier, and the protocol is zero-knowledge with respect to these parameters. Other parameters mentioned in a proof of knowledge expression are known to the verifier. (In particular, the description of the predicate W is known to the verifier.) For instance, $\mathrm{PK}\{(x, y) : W_1(x, y) = 1 \wedge W_2(x, z) = 1\}$ denotes a protocol where the parameters mentioned are (x, y, z); the value z is known to both parties (since it is not listed in parentheses); the protocol is zero-knowledge with respect to (x, y). Upon completion of this protocol, the verifier will be convinced that the prover knows some x' and y' such that $W_1(x', y')$ and $W_2(x', z)$ are satisfied.

Encryption Scheme

An *(asymmetric) encryption scheme* consists of the algorithms SetupEnc, Enc, and Dec with properties as follows.

- The *key-generation algorithm* SetupEnc outputs an encryption and decryption key pair (EK, DK).
- The *encryption algorithm* Enc takes as input a message m, a label L, and the encryption key EK and outputs an encryption E of m, i.e., $E = \mathrm{Enc}(m, L; EK)$.
- The *decryption algorithm* Dec takes as input an encryption E, a label L and the decryption key DK and outputs the message m, i.e., $m = \mathrm{Dec}(E, L; DK)$.

An encryption scheme is secure, if an encryption $E = \mathrm{Enc}(m, L; EK)$ does not contain any computational information about m to an adversary who is given E and EK, even if the adversary is allowed to interact with the decryptor. (For more on definitions of security for cryptosystems, see, for example, Goldreich [Gol04].) The notion of encryptions with labels was introduced in [CS98]. Labels allow one to bind some public data to the ciphertext at both encryption and decryption time. In our applications, the user would attach a label to an encryption E that indicates the conditions under which E may (should) be decrypted.

An encryption scheme that allows for efficiently proving in zero-knowledge that an encrypted value is the same as a value contained in a commitment or in a hidden certificate is called a *verifiable encryption scheme* [Bao00, Ate99, CD00, CS03].

Commitment Scheme

A *commitment scheme* consists of the algorithms Commit and VerifyCommit with properties as follows.

- The *commitment algorithm* Commit takes as input a message m and a random string r and outputs a commitment C, i.e., $C = \text{Commit}(m, r)$.
- The *(commitment) verification algorithm* VerifyCommit takes as input a C, m and r and outputs 1 (accept) if C is equal to $commit(m, r)$ and 0 (reject) otherwise.

The *security properties* of a commitment scheme are as follows. The *hiding property* is that a commitment $C = \text{Commit}(m, r)$ contains no (computational) information on m. The *binding property* is that given C, m, and r, where $1 = \text{VerifyCommit}(C, m, r)$, it is (computationally) impossible to find a message m' and a string r' such that $1 = \text{VerifyCommit}(C, m', r')$. Commitments form an important building block in systems based on zero-knowledge proofs.

Signature Scheme

A *signature scheme* consists of algorithms: SetupSign, Sign, VerifySig as follows:

- The *key-generation algorithm* SetupSign outputs a verification and signing key pair (VK, SK).
- The *signing algorithm* Sign takes as input a message m and a signing key SK and outputs a signature S on m, i.e., $S = \text{Sign}(m; SK)$.
- The *(signature) verification algorithm* VerifySig takes as input an alleged signature S, the message m, and the verification key VK; it decides whether to accept or reject the signature.

A signature scheme is secure [GMR88] if, on input VK, no adversary can produce a valid signature on *any* message m even after a series of adaptive queries to the signing algorithm (provided that the adversary did not explicitly ask for a signature on m).

To build selective disclosure certificates we use signatures with three special properties: (1) multi-block signing, (2) hidden signing, and (3) efficient proofs. Such signatures were first introduced in [CL03] and are often referred to as *CL-signatures*.

1. A conventional signature scheme usually hashes a big message into a group element that is then signed. For selective disclosure such an approach is

not viable. Often we do not want to destroy the semantic structure of attributes, e.g., the ordinality of a numbered attribute. Moreover we want to be able to access and selectively reveal each attribute on its own. In a CL-signature scheme Sign takes as input a *list* of messages $m_1, \ldots, m_l$ and a signing key SK and outputs a signature S on $m_1, \ldots, m_l$, i.e., $S = \mathrm{Sign}(m_1, \ldots, m_l; SK)$. The verification algorithm also looks at a list of messages and a purported signature.

2. For our purposes we also require a two-party protocol HiddenSign between a signer and a (signature) requestor. Let be given messages $m_1, \ldots, m_l$ and commitments $C_1 = \mathrm{Commit}(m_1), \ldots, C_{l'} = \mathrm{Commit}(m_{l'})$ with $l' \leq l$. The common inputs to the protocol are $C_1, \ldots, C_{l'}$ and $m_{l'+1}, \ldots, m_l$ and the signer's input is a signing key SK. At the end of the protocol the requestor's output is a signature S on $m_1, \ldots, m_l$. We denote such a protocol execution by $S = \mathrm{HiddenSign}(C_1, \ldots, C_{l'}, m_{l'+1}, \ldots, m_l; SK)$. We see that by the hiding property of commitments the signer does not learn any information on the messages $m_1, \ldots, m_{l'}$ in the protocol HiddenSign.

3. Finally, we require a protocol that allows the user (as the prover) to efficiently prove in zero-knowledge that she possesses a signature on committed messages. The following proof will form the core of the selective disclosure framework:

$$\mathrm{PK}\{(\sigma, m_1, \ldots, m_{l_1'}) :$$
$$\mathrm{VerifySig}(\sigma, m_1, \ldots, m_{l_1'}, m_{l_1'+1}, \ldots, m_{l_1}; VK) = 1\} \ .$$

10.3.3   Cryptography for the Controlled Release of Certified Data

In this section we discuss how the cryptographic building blocks discussed in the previous paragraph can be used to implement the controlled release of certified data.

By I_1 and I_2 we denote certificate issuers with verification and signing key pairs (VK_1, SK_1) and (VK_2, SK_2), respectively. The verification keys VK_1 and VK_2 shall be publicly known and authenticated. Also, we assume that the user holds a certificate $Cert_1 = \mathrm{Sign}(m_1, \ldots, m_{l_1}; SK_1)$ from I_1 and a certificate $Cert_2 = \mathrm{Sign}(\tilde{m}_1, \ldots, \tilde{m}_{l_2}; SK_2) = 1$ from I_2.

Multi-Show Unlinkability and Selective Show of Data Items

The key idea that underlies the controlled release of certified data is to prove knowledge (in zero-knowledge) of a certificate instead of disclosing a certificate to the verifier. To show the certificate $Cert_1$ to the verifier without disclosing, e.g., the data items $m_1, \ldots, m_{l_1'}$ (where $l_1' \leq l_1$), the user (as the prover) and the (certificate) verifier (as the verifier in the proof of knowledge) execute a protocol such as the following:

$$\text{PK}\{(Cert_1, m_1, \ldots, m_{l'_1}) :$$
$$\text{VerifySig}(Cert_1, m_1, \ldots, m_{l'_1}, m_{l'_1+1}, \ldots, m_{l_1}; VK_1) = 1\} \quad (10.1)$$

Protocol (10.1) proves that the user has (knows) a valid certificate with respect to the verification key VK_1. By the zero-knowledge property of the protocol, the verifier does not learn any information on $Cert_1$ and the data items $m_1, \ldots, m_{l'_1}$. From this observation it follows that multiple showings of the certificate $Cert_1$ using Protocol (10.1) are unlinkable, unless the data items $m_{l'_1+1}, \ldots, m_{l_1}$ disclosed to the verifier are linkable. The ability to selectively show data items follows trivially, as the user can choose, in each execution of Protocol (10.1), which data items to disclose to the verifier and of which data items to proof knowledge.

Proving Relations Between Data Items

This property is obtained in a straightforward way by using protocols such as the following

$$\text{PK}\{(Cert_1, m_1, \ldots, m_{l'_1}, Cert_2, \tilde{m}_2, \ldots, \tilde{m}_{l'_2}) :$$
$$\text{VerifySig}(Cert_1, m_1, \ldots, m_{l'_1}, m_{l'_1+1}, \ldots, m_{l_1}; VK_1) = 1$$
$$\wedge \text{VerifySig}(Cert_2, m_1, \tilde{m}_2, \ldots, \tilde{m}_{l'_2}, m_{l'_2+1}, \ldots, \tilde{m}_{l_2}; VK_2) = 1\} \ . \quad (10.2)$$

Using protocol (10.2) the user can prove that she possesses a certificate $Cert_1$ from I_1 and a certificate $Cert_2$ from I_2. Additionally, she proves that the first data items m_1 and $\tilde{m}_1$ of the certificates are equal. Yet, by the zero-knowledge property the verifier does not learn the respective data items. Thus we see that relations between certified attributes are demonstrated using zero-knowledge techniques to prove knowledge of relations, such as '=', '<', and '>'.

Conditional Showing of Data Items

Let us assume that there is a third party which, using the algorithm SetupEnc, has created the encryption and decryption key pair (EK, DK). The encryption key EK shall be publicly known and authenticated. To show, e.g., the data item m_1 contained in $Cert_1$ conditionally, the user encrypts m_1 under the encryption key EK of the third party, i.e., $E = \text{Enc}(m_1, Cond; EK)$. Here, $Cond$ denotes a label that describes the condition under which the user agrees m_1 to be released to the verifier. Then the user and the verifier execute the following protocol

$$\text{PK}\{(Cert_1, m_1, \ldots, m_{l'_1}) :$$
$$\text{VerifySig}(Cert_1, m_1, \ldots, m_{l'_1}, m_{l'_1+1}, \ldots, m_{l_1}; VK_1) = 1$$
$$\wedge E = \text{Enc}(m_1, Cond; EK)\} \ . \quad (10.3)$$

Besides of showing the certificate $Cert_1$, the user demonstrates in the protocol (10.3) that E is an encryption of the first data item contained in the certificate under the encryption key EK (such proofs are referred to as verifiable encryption). From the zero-knowledge property of the protocol and security property of the encryption scheme, it follows that the verifier does not get any (computational) information on the value encrypted in E.

To obtain the data item m_1, the verifier provides E and $Cond$ to the third party. The third party verifies whether the condition $Cond$ is fulfilled, and, if so, he returns the decryption $m_1 = \text{Dec}(E, Cond; DK)$ of E. We note that by the security property of the labeled encryption scheme, the third party cannot be fooled into decrypting under a condition other than the one described by $Cond$.

Blind Certification

Let us see how the user can get a certificate $Cert_3$ on data items m_1 and m' from issuer I_2 without disclosing m_1 to I_2, whereas the issuer I_2 can be asserted that m_1 is a data item certified by I_1; the data item m' is disclosed to the issuer. We recall that $Cert_1 = \text{Sign}(m_1, \ldots, m_{l_1}; SK_1)$. To this end, the user commits to m_1, i.e., $C = \text{Commit}(m_1, r)$. Then the user (as prover) and issuer (as verifier) execute the following protocol

$$\text{PK}\{(Cert_1, r, m_1, \ldots, m_{l'_1}): \quad C = \text{Commit}(m_1, r) \wedge$$
$$\text{VerifySig}(Cert_1, m_1, \ldots, m_{l'_1}, m_{l'_1+1}, \ldots, m_{l_1}; VK_1) = 1\} \ . \quad (10.4)$$

With this protocol the user demonstrates the issuer that C is a commitment to the first data item contained in the certificate $Cert_1$ issued by I_1. From the zero-knowledge property of the protocol and the hiding property of the commitment scheme, it follows that the issuer does not get any information on the data item m_1. If protocol (10.4) is accepted by the issuer, then he issues the certificate $Cert_3$ on m' and hidden m_1 using the protocol

$$Cert_3 = \text{HiddenSign}(C, m'; SK_2) \ , \quad (10.5)$$

where it is important to note that C is the same commitment as used in (10.4). From the properties of HiddenSign, it follows that in protocol (10.5) the issuer learns m' but does not learn any information on m_1.

Finally, the user checks the correctness of $Cert_3$ by evaluation of

$$\text{VerifySig}(Cert_3, m_1, m'; VK_2) = 1 \ .$$

The controlled disclosure techniques described above have a large number of applications to privacy protection, such as anonymous credential systems [Cha85a, CL00, Ver01, LRSW99], group signature schemes [CvH91, CS97a, ACJT00], and electronic cash [CFN90, Bra93].

10.4 Building Applications Using Idemix

In this section we sketch how one can use the above techniques to implement an anonymous credential system with anonymity revocation. We speak of anonymous credentials, if a user's selective disclosure certificates contain a unique and secret user identifier. Such an approach has several advantages: (1) Requiring all of the user's credentials to contain such an identifier is a powerful mechanism to avoid sharing of credentials. Even if users with a different identifier get full access to such credentials, they will not be able to show them together with their own credentials. (2) The secret identifier could be protected in a secure wallet such as a smart card. (3) Having a well-defined user identifier is a prerequisite for anonymity revocation.

We also show how the number and frequency of credential shows can be restricted using techniques based on e-cash with offline double-spending tests.

10.4.1 An Anonymous Credential System

The key idea underlying the implementation of anonymous credentials is that every user is represented by a unique identifier ID, which remains the user's secret throughout the lifetime of a credential system.

A credential from an organization simply is a certificate on the identifier ID (issued by the organization). Credentials are shown by using protocols of the form (10.1), such that the user's identifier ID is not disclosed to the verifier. Then the *unlinkability* of credentials follows from the (multi-show) unlinkability property of certificates discussed above. Credentials are issued using blind certification such that the user's ID is not disclosed to the issuing organization. The *unforgeability* of credentials trivially follows from the unforgeability property of the signature scheme being used for blind certification.

A credential system is called *consistent*, if it is impossible for different users to team up and to show some of their credentials to an organization and obtain a credential for one of them that a user alone would not have gotten [LRSW99, Lys02, CL01a]. We achieve consistency as follows. When the user shows multiple credentials from different organizations she proves that the same identifier ID underlies all credentials being shown, i.e., that the credentials belong to the same user. One possibility to achieve this is to use combined showing techniques as in protocol (10.2). Another possibility is the use of pseudonyms, as described below. When issuing credentials, the issuer asserts that the identifier ID it is blindly signing is the same as in existing credentials of the user. This can be achieved using the described blind certification protocols (10.4) and (10.5).

Attributes

Optionally, credentials can have attributes. Examples of credential attributes are an expiration date, the user's age, a credential subtype. When showing a

credential, the user can choose which attribute(s) to prove something about, and what to prove about them. E.g., when showing a credential that has attributes ($expdate = 2009/05/19$, $age = 55$), the user can decide to prove only that $age > 18$. Credential attributes are implemented by adding data items (additional to the user's identifier ID) to certificates. When showing credentials, the user can decide what information on attributes she discloses using the selective showing techniques described above.

A consistent and selective show of credentials can be expressed using the following protocol specification:

$$PK\{(Cert_{pass}, ID, birthdate, \ldots, Cert_{subscription}, expdate):$$
$$\text{VerifySig}(Cert_{pass}, ID, birthdate, \ldots; VK_1) = 1$$
$$\wedge \text{VerifySig}(Cert_{subscription}, ID, expdate; VK_2) = 1$$
$$\wedge today - birthdate < 25 \wedge today < expdate\} \ .$$

With this credential show, the user authenticates that he possesses a subscription, for example a train pass, that he is younger than 25, and that the subscription is not yet expired. The consistency of credentials guarantees that the user is not able to pass his subscription to a friend, unless he is also willing to share his passport credential.

Pseudonyms

Let us look in more detail at how credentials are issued. In (10.4) the user provides a commitment to the value (the user's identifier ID) that should be signed without being revealed to the issuer. For the subscription scenario above this would look like

$$PK\{(Cert_{pass}, ID, \ldots, r): \quad C = \text{Commit}(ID, r) \wedge$$
$$\text{VerifySig}(Cert_{pass}, ID, \ldots; VK_1) = 1\} \ .$$

In turn the user can obtain a subscription for the same hidden identity using (10.5):

$$Cert_{subscription} = \text{HiddenSign}(C, expdate; SK_2) \ .$$

In fact the organization can store C as a handle for this user and can issue new credentials to the user without repeating (10.4). It is convenient to speak of $C = Nym$ as the user's pseudonym. The simple and efficient proof $PK\{(ID, r): Nym = \text{Commit}(ID, r)\}$ is used to authenticate with respect to a pseudonym.

Using a pseudonym it is also possible to assure consistency between credentials that are not shown at the same time. Users can negotiate pseudonyms not only with issuers, but also with verifiers. Then users can show a credential

with respect to a pseudonym. The properties of the following show protocols assure that two credentials shown with respect to the same pseudonym are consistent:

$$\begin{aligned}
\mathrm{PK}\{(Cert_{pass}, ID, \dots, r') :\quad & Nym' = \mathrm{Commit}(ID, r') \wedge \\
& \mathrm{VerifySig}(Cert_{pass}, ID, \dots; VK_1) = 1\} \;, \\
\mathrm{PK}\{(Cert_{subscription}, ID, expdate, r') :\quad & Nym' = \mathrm{Commit}(ID, r') \wedge \\
& \mathrm{VerifySig}(Cert_{subscription}, ID, expdate; VK_2) = 1\} \;.
\end{aligned}$$

This mechanisms allows users to consecutively show more and more credentials under the same pseudonym in order to establish a long-term relationship with a verifier.

10.4.2 Anonymity Revocation

In many applications of credentials it is necessary that under certain conditions the user's anonymity is revoked. One can distinguish between global anonymity revocation in which a hash value $H(ID)$ of the hidden identifier ID that is common to all of a user's credentials is revealed, and local anonymity revocation in which only the pseudonym under which the credential was issued is revealed. Note that $H(ID)$ needs to be deposited together with information about a user's real-world identity at some organization. Often this organization is called the *credential root organization* and $H(ID)$ is deposited during the issuing of the so called *root credential*. It is not desirable that the credential root organization gets to know ID itself, as in that case it could impersonate honest users. Organizations that want to support global revocation must require a root credential show whenever they create a new pseudonym for a user.

Global anonymity revocation can be implemented in a straightforward fashion using our conditional showing techniques by conditionally disclosing the hash of the user's identifier ID during a show. The hash is verifiably encrypted using the revocation manager's public key EK, and tagged with a decryption condition.

For instance a car rental organization may require the following show of the renter's driver's license:

$$\begin{aligned}
\mathrm{PK}\{(Cert_{driverslicense}, ID, \dots) : \\
\mathrm{VerifySig}(Cert_{driverslicense}, ID, \dots; VK_1) = 1 \\
\wedge E = \mathrm{Enc}(H(ID), \text{'Serious accident'}; EK)\} \;.
\end{aligned}$$

If the car rental organization contacts the revocation manager with an encryption E, the revocation manager checks the available evidence. If he judges that the condition 'Serious accident' is fulfilled, he decrypts E and returns $H(ID)$. Now the car rental organization can ask the root organization for the real world identity of the user with $H(ID)$.

In order to allow for local anonymity revocation, a credential needs to be extended to include not only the identifier ID but also the randomness r of the commitment. First the user proves that the commitments are formed correctly:

$$\text{PK}\{(Cert_{pass}, ID, \ldots, r): \quad Nym = \text{Commit}(ID, r) \wedge C' = \text{Commit}(r, r') \wedge \\ \text{VerifySig}(Cert_{pass}, ID, \ldots; VK_1) = 1\} \ .$$

The extended credential is obtained by

$$Cert_{subscription} = \text{HiddenSign}(Nym, C', expdate; SK_2) \ .$$

For showing a credential that allows for local anonymity revocation, the user and the verifier run the following protocol:

$$\text{PK}\{(Cert_{subscription}, ID, r, \ldots, Nym): \\ \text{VerifySig}(Cert_{subscription}, ID, r, \ldots; VK_1) = 1 \\ \wedge Nym = \text{Commit}(ID, r) \\ \wedge E = \text{Enc}(Nym, \text{'Subscription shared on Internet'}; EK)\} \ .$$

Note that here the Nym is not revealed to the verifier. The user only proves that the pseudonym is encrypted in E and that it can be decrypted by the revocation manager owning the secret key corresponding to EK.

10.4.3 Balancing Anonymity and Accountability Using e-Cash Techniques

A simple offline anonymous e-coin scheme balances anonymity in the following way. The coin obtained from the bank can be spent anonymously, but only once. In this case even the bank cannot link the coin to the user that withdrew it. However, every additional spending of the coin reveals the identity of the user that tried to doublespend the coin. We want to leverage and generalize these techniques for anonymous credentials. We are interested in the following two functionalities: (1) restricting the number of credential shows, and (2) restricting the frequency of credential shows. As we will see, these two features are closely related.

First let us sketch an implementation of an anonymous e-cash system with offline double-spending tests based on the private certification framework in Section 10.3. Such a system consists of banks issuing e-coins, users spending e-coins at shops, which in turn deposit spent coins at the bank.

An e-coin is a certificate issued by the bank. To retrieve an e-coin, the user identifies herself at the bank. The bank assigns a unique number ID to the user. The user secretly chooses a random serial number S and a random blinding number b. The bank issues a certificate $Cert_{ecoin}$ on the data items ID, S, and b using blind certification such that it does not learn S and b.

At a shop the user spends the e-coin $Cert_{ecoin}$ as follows. The shop chooses a random integer challenge c. The user computes $U = ID \cdot c + b$ and uses the following variant of a selective showing protocol

$$\begin{aligned} \mathrm{PK}\{(Cert_{ecoin}, ID, b, ID', b') : \\ \mathrm{VerifySig}(Cert_{ecoin}, ID, s, b; VK) = 1 \\ \wedge u = (ID' \cdot c + b') \wedge ID = ID' \wedge b = b'\}, \quad (10.6) \end{aligned}$$

where VK is the bank's signature verification key. We note that the shop learns the value of s in the proof (10.6). Here we additionally assume that the proof (10.6) can be carried out non-interactively, i.e., it can be represented in terms of string Π which is sent from the user to the shop. Such a non-interactive proof can be validated by the shop by applying an appropriate verification algorithm on Π. Also, in analogy to the zero-knowledge property of interactive proofs, a non-interactive proof shall not reveal any (computational) information on $Cert_{ecoin}$, ID, and b. To deposit the e-coin, the shop sends the tuple (c, s, u, Π) to the bank. The bank first verifies the non-interactive proof Π to see if the tuple (c, s, u, Π) corresponds to a valid spending of an e-coin. In case of double spending the bank can recover the cheating user's ID as follows. The bank verifies if there already exists an e-coin with serial number s in its database of deposited e-coins. If so, it retrieves the corresponding tuple (c', s, u', Π'). We may safely assume that $c \neq c'$, and also we recall that by (10.6) the validity of Π asserts that $u = ID \cdot c + b$ and $u' = ID \cdot c' + b$. Therefore, from u, u', c, and c' the bank can compute the user's identity $ID = (u - u')/(c - c')$. Thus we see why non-interactive proofs are needed: it is because the bank itself needs to be able to verify the correctness of the proof (10.6) to ensure it correctly reveals a cheating user's identity ID.

Other desirable properties of e-cash, such as unforgeability and anonymity immediately follow from the properties of our certificates and the associated controlled disclosure techniques discussed above.

Extending Offline e-Cash

In order to generalize the above techniques to limiting the number of credential shows to $k > 1$ we introduce an additional cryptographic primitive.

A *pseudorandom function* is a function with a secret seed s, whose output cannot be distinguished from the output of a truly random function. We also require that an efficient zero-knowledge proof of knowledge can be made of the fact that a value y was computed as $y = \mathrm{PRF}(x; s)$.

The e-cash scheme above is now adapted. Instead of the serial number and the seed itself, the user secretly chooses a random serial number seed s and a random blinding value seed b. The bank issues a certificate $Cert_{cred}$ on the data items ID, s, b, and any other attributes that should be contained in the credential, e.g. a pseudonym, using blind certification such that it does not learn s and b.

At a verifier the user shows the certificate $Cert_{ecoin}$ and is required to follow the following procedure for some $1 \leq i \leq k$.

The verifier chooses a random integer challenge c. The user computes the serial number $S = PRF(i, s)$ and $U = ID \cdot c + PRF(i, b)$ and uses the following variant of a selective showing protocol

$$\begin{aligned}
\mathrm{PK}\{(Cert_{ecoin}, &ID, s, b, ID', s', b', i) : \\
&\mathrm{VerifySig}(Cert_{ecoin}, ID, s, b; VK) = 1 \\
&\wedge S = \mathrm{PRF}(i, s') \wedge U = (ID' \cdot c + \mathrm{PRF}(i, b')) \\
&\wedge ID = ID' \wedge s = s' \wedge b = b' \wedge 1 \leq i \leq k\},
\end{aligned} \qquad (10.7)$$

where VK is the verifier's signature verification key. The user is restricted to k shows, as she has only k possibilities of choosing i, and a reuse of an old i can be detected by S and allows for the extraction of ID using U.

Restricting the Frequency of Credential Shows

The frequency of credential shows can be restricted to k shows per time interval through the following simple extension to the above scheme. First, all users and verifiers need to agree on a (fixed-length) time interval identifier. For instance, they can choose to use the current date. Instead of computing S and U as described above, these values are now made to depend on the current time interval identifier t: $S = PRF(t\|i, s)$ and $U = ID \cdot c + PRF(t\|i, b)$. For a given t, a user has k possibilities to choose i, if she has used up all possible i values in one time period, the user has to wait until the verifier starts accepting credential shows for the next time interval.

10.4.4 Application Scenarios

We next discuss some application scenarios for our privacy-friendly authentication technology based on private certificates.

10.4.4.1 Passports: From the Airport to Second Life

The Idemix technology solves identity management in a way that applies to both physical real-world situations such as crossing borders between contries and virtual anonymous situations in video games. Let us explain: A traditional passport attempts to provide strong authentication in a physical setting. Passports are difficult to counterfeit (although not impossible), and they serve as a nation's most widely accepted form of certification of an individual's citizenship and identity. When passports are used to enter a country, typically all of the individual's information could potentially be recorded by the accepting country. A sovereign nation can provide its administration with the right to know the full identities of the people it allows to enter its borders. As we mentioned above, the Idemix technology, while an enabler of anonymity, is also

perfectly suited for this full-disclosure application. However, the interesting part of an Idemix solution is that, in addition to solving full-disclosure applications, the same passport can support varying levels of anonymity in other contexts. One such application at the very extreme end of anonymity comes from the popular role-playing game Second Life. Second Life gives users the opportunity to shape a whole electronic ecosystem according to their wishes. Put simply, Second life unites the capabilities of messaging, bulletin boards, 3-D environments and gaming, and all sorts of content generation. However, the new forms of interactions which emerge from Second Life are not just purely for fun; indeed, several companies such as Adidas, Reebok, 20th Century Fox, Intel, and IBM have actually created a commercial presence in this virtual world. As one can imagine, building trust in this virtual world becomes an even harder problem than building trust in the physical one. The question of identity becomes blurred because one of the features of Second Life is the ability for its users to have a "second life" in which their identity is like a blank canvas. The solution we envision involves the use of Idemix certified attributes from both the real world as well as the Second Life one. Indeed, credentials will prove valuable to the user holding them and to all user's interacting with that user. Coming back to the running example in this document, an Idemix passport credential issued by a nation can be used to allow only teenagers to enter a virtual teenager island in this virtual world by proving that they hold a real-world electronic passport credential certifying their age. These types of proofs guarantee anonymity in the virtual world, i.e., they exactly preserve one of the virtues of Second Life, while simultaneously making it a safer place, free from internet predators and fraudsters.

Such applications often require to prove membership in a large group of people, e.g., all teenagers, or all passport holders that are members of the European Union. In fact it may be desirable to combine the two restrictions: admitting teenagers from different countries, while restricting the set of admissible countries. There are different ways for proving that a value is in an interval or in a set. To show the power of our certification framework we show a technique that is based solely on selective disclosure certificates.

In a preparatory step, the verifier publishes a certificate for each set element: $Cert_{10}, \ldots, Cert_{19}$ and $Cert_A, \ldots, Cert_{UK}$.

When showing her credential the user now also proves that her attribute value corresponds to one of these values:

$$
\begin{aligned}
\mathrm{PK}\{ & (Cert_{cred}, Cert_{age}, Cert_{country}, ID, age, country): \\
& \mathrm{VerifySig}(Cert_{cred}, ID, age, country; VK) = 1 \\
& \wedge \mathrm{VerifySig}(Cert_{age}, ID, age, \text{'teenager'}; VK_{age}) = 1 \\
& \wedge \mathrm{VerifySig}(Cert_{country}, ID, country, \text{'EU'}; VK_{country}) = 1\}, \quad (10.8)
\end{aligned}
$$

By proving that her *age* attribute values correspond to one of the certificates with the *teenager* tag, and that her *country* attribute value corresponds

to one of the values with the *EU* tag, the user proves that she fulfills the admission criteria.

This example also shows a restriction of the current idemix system. In order to hide the country of the user, all passport credentials would need to be issued with the same issuing key. Something that is unlikely to happen as countries try to protect their sovereignty. This restriction can be overcome by delegatable credentials. Using delegatable credentials, an international organization like the OECD or the United Nations could form the root of a certification hierarchy, and would allow member states to sign passports for their citizens. When using such credentials, users could prove that they have a valid EU passport, while hiding from which member state it was obtained.

10.4.4.2 Web 2.0: User Content Generation

Given the immense popularity of blogs, wikis, wikipedias, and sites such as YouTube, it seems clear that Web 2.0 is about content generated by users. The quality of such user-generated content may vary considerably. Even more importantly, manipulations of content such as recent examples of wikipedia manipulations to influence US congress elections can be particularly harmful to both individuals and a society that increasingly relies on such information. Of course, currently we are only experiencing the first outbursts of such attacks and they will grow unless we act quickly to build trust into content generation. As a first observation, we note that the value of user-generated content strongly depends on the source's reputation and attributes. Content that was generated by users with certified attributes will be easier to trust and more acceptable to a global community. As an example, a medical statement about a disease will be perceived quite differently in the cases of it being made by some anonymous user or by a specialist in the field. Certainly, users of Web 2.0 still want to benefit from the ability to create content anonymously and free of coercion. Thus, we see Idemix and true user-centric identity management as an enabler of more secure and reliable user content-generation for Web 2.0.

10.4.4.3 Privacy in Health Care Insurance

Health care is one of the most privacy-sensitive areas in the digital realm. But even in the physical realm, health-care systems require even more sophisticated and robust privacy protection mechanisms. Consider, as an example, an insurance company that would like to offer its clients a free method to take sensitive disease screenings (e.g, sexually-transmitted diseases) periodically throughout the year. The insurance company has an ethical interest in improving the user's health by offering these complementary tests, however, it must also contend with users' concerns about the privacy of the test results, and moreover, even the privacy infringements relating to how the client submits reimbursement claims for the test. One might consider, at first, a

completely anonymous testing service. Such a system, however, might expose the insurance company to abuse by clients who take the tests too often, or non-clients who nonetheless take tests at the insurance company's expense. (Notice, at the core level, this problem is one of accountability.) The PRIME solution to true user-centric identity management address both of these problems at once. In one instantiation, clients can be issued smartcards by the insurance company which contains details about their plan. This insurance card can automatically generate a credential allowing the user to receive a test up to k times a year. Thus, the test can be done untraceably with full anonymity and the payment can be processed in bulk by the insurance company. If the user tries do the test more than k-times a year (or tries to sharing the credential with user's not entitled to the insurance protection), her identity will be revealed, and she can be billed for the excess tests.

10.5   Historical Notes

As many other privacy-enabling protocols, Chaum put forth the principles of anonymous (or private) credentials [Cha85a, CE87]. Later, Damgård gave the first proof of concept [Dam90] of an anonymous credential system where a credential was represented by a signature on an individual's name, obtained in a way that kept the name hidden from the issuer while showing a credential was carried out via a general purpose zero-knowledge proof of knowledge. Due to the practical inefficiency of these zero-knowledge proofs, this first solution was rather of theoretical interest.

The first step towards efficient systems was Brands e-cash schemes and protocols to issue signature on hidden message. Brands later put these building blocks together to build a private credential system [Bra95, Bra99]. The first truly efficient and provably secure scheme was put forth by Camenisch and Lysyanskaya [CL01a], whose construction was largely inspired by the Ateniese et al. group signature scheme construction [ACJT00]. Their system allowed users for the first time to use a credential more than once.

Today, the probably most prominent real application is the Direct Anonymous Attestation [BCC04] protocol employed by the Trusted Computing group to authenticate a trustworthy computing platform while retaining the user's privacy.

11

Privacy Models and Languages: Access Control and Data Handling Policies

Claudio Agostino Ardagna, Sabrina De Capitani di Vimercati,
and Pierangela Samarati

Università degli Studi di Milano

11.1 Introduction

The huge amount of personal information available on the Web has led to growing concerns about the privacy of its users, which has been recognized as one of the main reasons that prevents users from using the Internet for accessing online services. Users in fact prefer not to be under the control of anyone at anytime. In this context, the concept of *privacy control* is introduced, and it should encompass three main aspects: to guarantee the desired level of privacy of information by controlling the access to services/resources; to control secondary use of information disclosed for the purpose of access control enforcement; to deal with the specific management of related privacy obligations [Cas04b] (e.g., data retention, data deletion, notifications).

In this privacy-oriented scenario, access control systems may help users in keeping control over their personal information. Access control solutions should then be enriched with the ability of supporting privacy requirements [ADDS05, BDDS01], as for instance: *i) interchangeable policy format,* parties need to specify protection requirements on the data they make available using a format both human- and machine-readable, easy to inspect and interchange; *ii) interactive enforcement,* the evaluation phase should provide a way of interactively applying criteria to retrieve the correct reports, possibly managing complex user interactions, such as, the acceptance of written agreements and/or online payments for each report; *iii) metadata support,* privacy-aware access control systems should allow to specify access restrictions based on conditions on metadata describing (meta)properties of the stored data and the users.

Traditional access control systems, which are based on regulations (*policies*) that establish who can, or cannot, execute which actions on which resources [SD01], result limiting and do not satisfy the above requirements. Although recent enhancements allow the specification of policies with reference to generic attributes/properties of the parties and the resources involved (e.g., XACML [eXt05]), access control systems are not designed for enforcing privacy policies. Also, few proposals have tried to address the problem of how to regulate the use of personal information in secondary applications. The consideration of privacy issues introduces the need for rethinking authorization policies and models, and the development of new paradigms for access control policy specification and enforcement. Two main issues to be looked at are:

1. access control needs to operate even when interacting parties wish to remain anonymous or to disclose only specific attributes about themselves;
2. data collected/released during access control, as well as data stored by the different parties, may contain sensitive information on which privacy policies need to be applied.

In the following of this chapter, we will provide further details about different types of privacy policies managed in PRIME. Chapters 12 and 13 will then investigate more in detail privacy obligations and assurance policies, respectively.

11.2   Privacy Policy Categories

To fully address the requirements introduced by the need of a privacy-aware access control system, a new model together with the following different types of privacy policies have been introduced.

Access Control Policies

Access control policies define authorization rules concerning access to data or services [SD01]. Authorizations correspond to traditional (positive) rules usually enforced in access control systems. For instance, an authorization rule can require a user of age and a credit card number (condition) to read (action) a specific set of data (object). When an access request is submitted to a service provider, it is evaluated against the authorization rules applicable to it. If the conditions for the required access are evaluated to true, access is permitted. If none of the specified conditions that might grant the requested access can be fulfilled, access is denied. Finally, if the current information is insufficient to determine whether the access request can be granted or denied, additional information is needed, and the requester receives an undefined response with a list of requests that she must fulfill to gain the access. For instance, if some of the specified conditions can be fulfilled by signing an agreement, then the party prompts the requester with the actions that would result in the required access.

Release Policies

Release policies define the preferences of each party regarding the release of its PII. They specify to which party, for which purpose/action, and under which conditions a particular set of PII can be released [BS02a]. For instance, a release policy can state that credit card information can be released only in the process of a purchase and to trusted partners. The release of PII may only be enforced if the release policies are satisfied.

Data Handling Policies and Obligations

Data handling policies [ADS06, ACDS08] regulate how PII will be handled at the receiving parties (e.g., information collected through an online service may be combined with information gathered by other services for commercial purposes). Users specify these policies to define restrictions on secondary use of their personal information, thus controlling the information also after its release. Data handling policies will be attached to the PII or data they protect, and transferred as *sticky policies* to the counterparts [KSW02b]. A specific type of data handling policy is the obligation policy (see Chapter 12) dictating privacy constraints and expectations on the lifecycle management of personal data. For example, these policies might prescribe constraints on data deletion, data transformation, notifications, and the like.

Assurance Policies

Assurance policies describe enterprise assurance properties relating to how personal data will be transferred, processed, and protected (see Chapter 13). They can help in checking compliance to law, data subjects' preferences and enterprise guidelines, and can refer to trust, assurance, and contextual properties.

The next sections briefly describe a scenario illustrating few examples of policies that have been managed in PRIME, focusing on access control and data handling policies. Obligation and assurance policies will be discussed more in detail in the following chapters.

11.3   Scenario

Our reference scenario is a distributed infrastructure that includes three parties (see Figure 11.1):

- *users* are human entities that request online services;
- *service provider* is the entity that provides online services to the users and collects personal information before granting an access to its services;
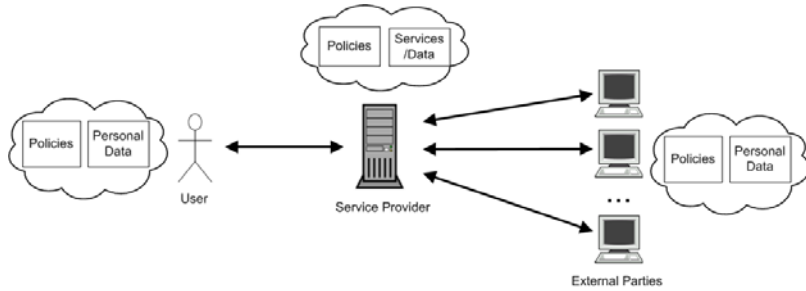
Fig. 11.1 Reference scenario

- *external parties* are entities (e.g., business partners) to which the service provider may want to send or trade personal information of users. Alternatively, they may be involved in the process of checking and providing evidence that the service provider's assurance policies are valid.

Although each party can act interchangeably as a user, a service provider, or an external party during different transactions, they usually have well defined, fixed roles when one specific access request is considered. We assume that the service provider collects personal data that are necessary to provide access to services and stores them into *profiles* associated with each user. A profile can therefore be seen as a container of pairs of the form $\langle attribute\_name, attribute\_value \rangle$, where $attribute\_name$ is the name of the attribute provided by the user and $attribute\_value$ is its value.

The set of messages exchanged between a user and a service provider is called *negotiation* [BS02a]. A negotiation always starts with a user request to a service provider and ends explicitly (done or stop) or implicitly, for example, assumed after a certain timeout period. A negotiation intuitively corresponds to the classical concept of *session*. We assume anonymous communications to be in place. Therefore, at the beginning of a negotiation, the user is unknown to the service provider, meaning that no information about the user has been collected by the service provider. During a negotiation a user may, similarly to a service provider, require the counterpart to fulfill some requirements (i.e., provide information) for it to proceed. In others words, a bidirectional negotiation is considered where both parties can require the other party to provide them with certain information or digital certificates necessary for service request or fulfillment. Each party has a *portfolio of credentials* (third-party endorsed attribute data such as digital certificates) and *declarations* (unsigned data). Access to services and release of portfolio information are managed according to the rules specified by the parties.

The same discussion is still valid when an external party wants to access personal information of the users stored at the service provider. The only differences are that, in this case, the service provider must be responsible for protecting the privacy of users' data, and users are assumed to trust the

service provider to faithfully maintain and manage their personal information, according to their privacy requirements. The process of negotiation enables users (or entities acting on their behalf) to have a better understanding of which type of data is going to be required to fulfill a transaction, for which purposes, and under which constraints.

In the PRIME vision, the negotiation and PII disclosure phases should also enable users to specify their privacy preferences in terms of data handling and management. Ultimately, these preferences dictate constraints and obligations to be fulfilled by the data receiving party. This scenario has implications on how to represent these preferences and how to factor them into privacy-aware obligation policies. PRIME also wants to enable users to check upfront the properties and capabilities of a data receiving party (e.g., an organization, a service provider, and so forth), before engaging in any data disclosure. This is part of the 'assurance checking' process, aiming at increasing the level of trust a user has in an organization.

In the remainder of this chapter and in the next two chapters, we illustrate the basic concepts and principles of our models for access control, obligation and assurance policies.

11.4 Access Control Model and Language

An access control language is used to specify both access control and release policies.[1] In this section, we give an overview of the functionalities and syntax provided by our access control model and language.

11.4.1 Basic Concepts

11.4.1.1 Portfolio and Profiles

In open environments, the decision to grant access to a resource is often based on different attributes of the requester rather than its specific identity. Here, we assume that each party has a *portfolio* of *declarations* and *credentials* [GEB], which is used to gain (or offer) services [BS02a]. The portfolio may also represent views of certificates that are not actually stored at the party site but can be obtained if needed [SAB$^+$]. This way, the model allows a party to refer to the set of all its possible credentials without need of maintaining a copy of each of them. The definition of portfolio introduces the following types of attributes [BS02a].

- *Certified attributes* are specified in an electronic credential that is characterized by the credential *name*, the *issuer*'s public key, the *subject*'s public

---

[1] Although semantically different, access control and release policies are syntactically identical.

key, a *validity period*, a list (possibly empty) of pairs ⟨*attribute_name*, *attribute_value*⟩ representing the certified attributes (e.g., name and surname contained in an electronic passport), and a *digital signature*.

- *Declared attributes* represent self-certified statements of a party, with no certification from any legal authority. A declared attribute is a pair ⟨*attribute_name*, *attribute_value*⟩ (e.g., the professional status of a user communicated by the user herself).

The set of certified and declared attributes released by a party to the service provider is then stored in the profile associated with the party. To define restrictions or to identify a party based on its attributes, we introduce the concepts of *credential term* and *declaration term*. Let $\mathcal{C}$ be a set of credential names and $\mathcal{Q}$ a set of predicates including standard built-in mathematical predicates (e.g., `equal`, `notEqual`, `greaterThan`). We define a credential term as follows.

Definition 1 (Credential term). Given a credential name `cred_name` $\in \mathcal{C}$, a *credential term* over `cred_name` is an expression of the form `cred_name`(*condition_list*), where *condition_list* is a list of expressions of the form `math-pred`(*attribute_name*,*value*) with `math-pred` $\in \mathcal{Q}$ a mathematical predicate, *attribute_name* the attribute name as it appears in the credential `cred_name`, and *value* the corresponding attribute value.

Expression *condition_list* permits to define a list of conditions that are treated as if ANDed, and that allow to define restrictions on a single credential without introducing variables in the language. We then define a binary predicate `credential`(*ct*,*K*), where *ct* is a credential term `cred_name`(*condition_list*), and *K* is the public key or the name of a trusted authority. Predicate `credential` is evaluated to true if and only if there exists a credential `cred_name` issued by an authority *K* and such that *condition_list* is satisfied.

Example 1. An example of credential term is `identity-card`(`equal`(occupation,'Student')) denoting an `identity-card` credential whose attribute occupation has value Student. Predicate `credential`(`identity-card`(`equal`(occupation,'Student')),K_1) is then evaluated to true if there exists an `identity-card` credential issued by K_1 certifying that the occupation of the credential subject is Student.

A declaration term is defined as follow.

Definition 2 (Declaration term). A *declaration term* is an expression of the form `predicate_name`(*arguments*), where `predicate_name` $\in \mathcal{Q}$ is the name of a generic predicate, and *arguments* is a list, possible empty, of constants or attributes.

We define a unary predicate `declaration`(*d*), where *d* is a declaration term `predicate_name`(*arguments*). Predicate `declaration` is evaluated to true if

and only if there exists a set of attributes such that
`predicate_name`(*arguments*) is satisfied.

Example 2. An example of declaration term is `equal`(name,'Alice') denoting
the name attribute whose value is Alice. The corresponding declaration pred-
icate is then `declaration`(`equal`(name,'Alice')).

Declarations and credentials in a portfolio may be organized into a partial
order. For instance, an `identity-document` can be seen as an abstraction for
`driver-license`, `passport`, and `identity-card`.

11.4.1.2 Ontologies and Abstractions

Our model provides the support for *ontologies* that permit to make generic
assertions on subjects and objects [DDFS04]. More precisely, we use three
ontologies: a *subject* ontology, an *object* ontology, and a *credential* ontology. A
subject ontology contains terms that can be used to make generic assertions on
subjects and to define relationships among them. An object ontology contains
domain-specific terms that are used to describe resource content. Finally, a
credential ontology represents relationships among attributes and credentials
(`part-of` and `is-a` relationships), and more complex relationships between
attributes and abstractions. The credential ontology is then used to establish
which credentials can be provided to fulfill a declaration or credential request,
according to the principle of the *minimum disclosure.*

11.4.2 Functionalities

Before presenting the access control model and language used to specify the
access control policies protecting server-side resources and the *release policies*
regulating access to personal information of the parties, we summarize their
main functionalities as follows.

- *Attribute-based restrictions.* The language supports the definition of pow-
  erful and expressive policies based on properties (attributes) associated
  with subjects (e.g., name, address, occupation) and objects (e.g., owner,
  creation date). The language includes some operators for comparing at-
  tribute values and could be extended by adding nonstandard functions.
- *XML-based syntax.* The language provides an XML-based syntax for the
  definition of powerful and interoperable access control and release policies.
- *Credential definition and integration.* The language supports requests for
  certified data, issued and signed by authorities trusted for making the
  statement, and uncertified data, signed by the owner itself.
- *Anonymous credentials support.* The language supports definition of con-
  ditions that can be satisfied by means of zero-knowledge proof [CL01c,
  CV02].

- *Support for context-based conditions.* The language allows the definition of conditions based on context information (including the physical position of the users [ACD+06]). It further integrates metadata identifying and possibly describing entities of interest, such as subjects and objects, as well as any ambient parameters concerning the technological and cultural environment where a transaction takes place.
- *Ontology integration.* Policy definition is fully integrated with subject and object ontologies in defining access control restrictions. Also, the language takes advantage of the integration with a credential ontology that represents relationships among attributes and credentials, and between credentials.

11.4.3 Description of the Access Control Language

We describe our access control model discussing the basic constructs of the language used to define access control and release policies. First, the following predicates constitute the basic literals that can be used in access control and release policy specification:

- a binary predicate `credential(`*ct*`,`*K*`)`, where *ct* is a *credential term* (see Definition 1), and *K* is the name or the public key of a trusted certification authority (CA);
- a predicate `declaration(`*d*`)`, where *d* is a list of *declaration term* (see Definition 2);
- a set of standard built-in mathematical predicates, such as `equal()`, `greater_than()`, `lesser_than()`, and so forth;
- a set of state-based, location-based, trust-based predicates of the form `predicate_name(`*arguments*`)`;
- a set of non-predefined predicates.

Then, three basic elements of the language have been identified: *subject_expression*, *object_expression*, and *conditions*. Below, single properties belonging to user and object profiles are referenced through the dot notation. Also, to refer to the requester (i.e., the subject) and the target (i.e., the object) of the request being evaluated without the need of introducing variables in the language, we use keywords **user** and **object**, respectively, whose appearances in a conditional expression are intended to be substituted with actual request parameters during run-time evaluation of the access control policy. For instance, `Alice.Address` indicates the address of user `Alice`. Here, `Alice` is the pseudonym of the user (and therefore the identifier for the corresponding profile), and `Address` is the name of the property. Beside the formal definition of the access control model and language, we also provide some examples of *subject_expression*, *object_expression*, and *conditions* using the XML-based syntax defined for access control and release policy specification. This syntax has been used in the development of our privacy-aware access control system prototype shown in Section 14.

Subject Expression

These expressions allow to refer to a set of subjects depending on whether they satisfy given conditions that can be evaluated on the subject's profile. More precisely, *a subject expression is a boolean formula of* `credential` and `declaration` (see Definition 1 and 2). The following are examples of subject expressions:

- `credential(passport(equal(`**user**`.nationality,'Italian'),` `greater_than(`**user**`.age,18)),`K_1`)` denoting requests made by Italian users of age. These properties should be certified by showing the `passport` credential verifiable with public key, or released by CA, K_1;
- `declaration(equal(`**user**`.name,'John'))` denoting requests made by a user whose name is John.

Based on the syntax provided in Appendix 30.1, an example of *subject expression* is provided in the following where *any* user[2] must provide her name (i.e., *name.given*) and surname (i.e., *name.last*) proved by an X.509 *identity-document* released by the Italian public administration, and must be of age (no certification is requested) to gain the access to a particular object.

```
<subject>any</subject>
<subjectExprs>
 <group>
   <condition name="exist">
     <argument isLiteral="false">name.given</argument>
   </condition>
   <condition name="exist">
     <argument isLiteral="false">name.last</argument>
   </condition>
   <evidence>
     <issuer>ItalianPublicAdministration</issuer>
     <proofMethod>X.509</proofMethod>
     <type>identity-document</type>
   </evidence>
 </group>
 <group>
   <condition name="greaterThan">
     <argument isLiteral="false">age</argument>
     <argument isLiteral="true">18</argument>
   </condition>
   <evidence/>
 </group>
</subjectExprs>
```

---

[2] The `subject` element defines an identifier or an abstraction that refer to a set of users.

The **group** element groups different conditions that have to be satisfied by the same **evidence** element (i.e., the same credential), which in turn defines restrictions on the certification type. This avoids that a request for *name.given* and *name.last* is satisfied by two different credentials.

Object Expression

These expressions allow to refer to a set of objects depending on whether they satisfy given conditions that can be evaluated on the object's profile. More precisely, *an object expression is a boolean formula of terms of the form* **predicate_name(***arguments***)**, where *arguments* is a list, possible empty, of constants or attributes. The following are examples of object expressions:

- **equal(object**.owner,**user)** denoting all objects created by the requester;
- **lesserThan(object**.validity,today) denoting all valid objects;
- **greaterThan(object**.age,35) denoting all objects whose attribute age is greater than 35.

Based on the syntax provided in Appendix 30.1, an example of *object expression* is provided in the following, specifying the set of all credit cards (i.e., *cc_info*)[3] with VISA circuit and whose expiration date was before December 2000.

```
<object>cc_info</object>
<objectExprs>
 <condition name="equal">
   <argument isLiteral="false">circuit</argument>
   <argument isLiteral="true">VISA</argument>
 </condition>
 <condition name="lesserThan">
   <argument isLiteral="false">expiration</argument>
   <argument isLiteral="true">12/00</argument>
 </condition>
</objectExprs>
```

Conditions

Conditions element specifies conditions that can be brought to satisfactions at run-time processing of the request. More precisely, *a condition element is a boolean formula of terms of the form* **predicate_name(***arguments***)**, where *arguments* is a list, possible empty, of constants or attributes. Four different types of conditions can be stated inside a rule: *i) state-based conditions*: restrictions based on the environment state; *ii) location-based conditions*: restrictions based on location information of individuals; *iii) trust-based conditions*: restrictions based on the assurance/trust of the environment; *iv) others conditions*: conditions that do not belong to any of the other classes.

---

[3] **object** element defines an object identifier or abstraction.

Each condition type is defined by means of an ad-hoc XML element (see Appendix 30.1): `stateExprs`, `lbsExprs`, `trustExprs`, `genericExprs`. In the following, we provide an example of location-based condition stating that, at access control time, the country under which the roaming phone of the requester (i.e., *SIM*) is registered should be 'Italy' to have the request satisfied.

```
<lbsExprs>
 <condition name="equal">
   <argument isLiteral="false">SIM</argument>
   <argument isLiteral="true">Italy</argument>
 </condition>
</lbsExprs>
```

The same syntax of `lbsExprs` element is used for all types of conditions.

11.4.3.1 Policy and Rule Definition

An access control policy (release policy, resp.) is composed by one or more rules, composed in OR logic between them, directly associated with an object component and the related set of actions. Syntactically, an access control policy (release policy, resp.) can be formalized as follows.

Definition 3 (Access control policy). *An access control policy is an expression of the form* ⟨actions⟩ *ON* ⟨object⟩ WITH ⟨object_expression⟩ IF ⟨rules⟩, *where:*

- actions *is the set of actions to which the rules refer (e.g., read, write, and so on)*.[4]
- object *identifies the object to which the rules refer and corresponds to an object identifier or a named abstraction of values, if abstractions are defined on objects;*
- object_expression *is a boolean expression that allows the reference to a set of objects depending on whether they satisfy given conditions that can be evaluated on the object's profile;*
- rules *is a set of rules as defined in Definition 4.*

An access control rule (release rule, resp.) represents the basic element used to regulate the access to the objects with which it is associated. Syntactically, an access control rule (release rule, resp.) can be formalized as follows.

Definition 4 (Access control rule). *An access control rule is an expression of the form* ⟨subject⟩ WITH ⟨subject_expression⟩ CAN ⟨actions⟩ FOR ⟨purposes⟩ IF ⟨conditions⟩, *where:*

---

[4] Note that the *actions* field can be refined in the rules. Abstractions can also be defined on actions, specializing actions or grouping them in sets.

- subject *identifies the subject to which the rule refers and corresponds to a user identifier or a named abstraction of values, if abstractions are defined on subjects;*
- subject_expression *is a boolean expression that allows the reference to a set of subjects depending on whether they satisfy given conditions that can be evaluated on the user's profile;*
- actions *is the set of actions to which the rule refers (e.g., read, write, and so on);*
- purposes *is the purpose or a group thereof to which the rule refers, and represents how the data are going to be used by the recipient;*
- conditions *is a boolean expression of conditions that an access request to which the rule applies has to satisfy.*

Example 3. In the following, for sake of clarity and conciseness, access control and release policies are provided in the simplified form shown in Table 11.1, rather than with our complete XML-based syntax (see Appendix 30.1). As an example, suppose that a `hospital` provides a set of services to its patients. Patients release their data to the `hospital` to gain access to the services. The `hospital` defines the access control policies in Table 11.1 to protect the access to the data stored locally. In particular, AC1 is composed by two rules that regulate access to valid `cc_info`. An access control policy is evaluated to true if at least one rule is satisfied, that is *subject_expression* and *conditions* of the rule are satisfied. AC2 is composed by a single rule that regulates access to `personal_info` of patients.

To conclude, although the definition of access control and release policies permits to protect access to data and services, and release of personal data, respectively, no solution is provided for regulating how PII must be used and processed after its release. To this aim, in the next section, we introduce a data handling model and language.

11.5 Data Handling Model and Language

A privacy-aware access control solution supporting restrictions on secondary use should be simple and expressive enough to support, among others, the following privacy requirements [Dir95, Org80]: *i) openness,* privacy practices should be transparent and fully understandable for all parties; *ii) individual control,* users should be able to specify who can see what information about them and when; *iii) collection limitation,* parties collecting personal data for the purpose of a transaction must gather no more data than what is strictly needed; *iv) purpose specification,* entities who collect and disseminate personal data must specify the purposes for which they need these data; *v) consent,* users should be able to give their explicit and informed consent on how to use their personal data.

Table 11.1 An example of access control policies (release policies, resp.)

	object	act	AC Rules	Description
			Access Control Policies	
AC1	cc_info WITH greaterThan(**object**.expiration, *today*)	read	any WITH [credential(employeeCard(equal(**user**.job,'Secretary')),K_H) AND declaration(equal(**user**.company, 'Hospital'))] CAN read FOR service_release IF [in_area(**user**.sim,'Hospital') AND log_access()]	The secretaries of the Hospital are authorized to read valid (i.e., not yet expired) cc_info for service release purpose, if they are located inside the hospital and access is logged.
			any WITH [credential(employeeCard(equal(**user**.job,'BusinessConsultant')), K_H)] CAN read FOR reimbursement	The business consultants of the Hospital are authorized to read valid cc_info for reimbursement purpose.
AC2	personal_info WITH equal(**object**.physicianID, **user**.ID)	read	any WITH [credential(employeeCard(equal(**user**.job,'Primary Physician'), K_H) AND declaration(equal(**user**.company,'Hospital'))] CAN read FOR service_release	A primary physician of the hospital can read personal_info of her patients for service release purpose.

Our privacy-aware access control solution is based on *data handling policies* [ACDS08, ADS06] (DHPs, for short), respectful of the above requirements, which provide the users with the possibility to define how their PII can be subsequently used by the service provider and/or external parties. In the data handling policy specification, two issues need to be discussed: *by whom* and *how* a policy is defined. With respect to the first issue (i.e., by whom a DHP is defined), three different strategies are possible, each one requiring different levels of negotiation between a user and a service provider: *server-side*, *user-side*, and *customized*. Server-side and user-side are the opposite endpoints of all possible approaches in the definition of privacy rules that balance between service provider and user needs. The customized approach, instead, represents a trade-off between the power given to the service providers and the protection assured to the users. In particular, when a user requires a service, a predefined policy template is provided by the service provider as a starting point for creating data handling policies. The template is customized by the user to meet different privacy requirements. A user can directly customize the template or it can be supported by a customization process that automatically applies the privacy preferences of the user. If the customized data handling policy will be accepted by the service provider, the personal information provided by the user will be labeled with this policy. This represents the most flexible and balanced strategy for the definition of a data handling policy, and we therefore adopt it.

With respect to the second issue (i.e., how a DHP is defined), data handling policies are defined as independent rules and represent the privacy preferences of the users. DHPs should then include different components that allow users to define how the external parties can use their personal data. Personal data are *tagged* with such data handling policies. Syntactically, access control policies and data handling policies are similar, since a data handling policy regulates which *subject* can execute which *actions* on which *resources* under which *conditions* and following some *obligations*. Although the stand-alone option can introduce some redundancy in policy definition, it provides a better separation between policies that are used with two different purposes. This clear separation makes data handling policies more intuitive and user-friendly, and implicitly suggests the differences with access control policies. Also, the definition of standalone policies reduces the risks of unprotected data types and allows for the customization of additional components such as recipients and actions. Finally, an additional motivation to prefer stand-alone data handling policies is that some of the conditions (e.g., some obligations) do not necessarily depend on access control events, and then cannot just be enforced by an access control system. For instance, the obligation condition "delete data after 10 days" is enforced independently from the fact that the data have ever been accessed. In this case, stand-alone data handling policies enable building a solution with multiple enforcement points, some of them outside the control of the access control system (e.g., an obligation manager), but orchestrated/configured by it.

In the following, we assume a customized stand-alone approach for data handling policy specification.

11.5.1   Description of the Data Handling Language

The following predicates constitute the basic literals that can be used in data handling policy specification:

- a binary predicate `credential(`*ct*`,K)`, where *ct* is a *credential term* (see Definition 1), and K is the name or the public key of a trusted certification authority (CA);
- a predicate `declaration(`*d*`)`, where *d* is a list of *declaration term* (see Definition 2);
- a set of standard built-in mathematic predicates, such as `equal()`, `greater_than()`, `lesser_than()`, and so forth;
- a set of provision and obligation predicates of the form `predicate_name(`*arguments*`)`;
- a set of non predefined predicates.

Five basic elements have then been identified: *recipients*, *purposes*, *PII abstraction*, *restrictions*, and *obligations*. As for our access control language, we provide an XML-based syntax for data handling policy specification (see Appendix 30.2).

Recipients

A recipient is an external party to which PII of users can be disclosed by the service provider [Dir95]. Since external parties may be unknown to the user, she should define to which entities her data may be disclosed without knowing their identity. Our approach supports the definition of recipients based on their *attributes*, instead of their *identity*. Similarly to *subject_expression* in access control policies, *recipients* is a boolean formula of `credential` (see Definition 1) and `declaration` (see Definition 2).

   Based on the syntax provided in Appendix 30.2, an example for a *recipients* element is provided in the following where an external party can access user data provided that it belongs to the *Public Administration* or it is a *Non-profit Organization*.

```
<recipients>
 <recipient>
   <condition name="equal">
     <argument isLiteral="false">type</argument>
     <argument isLiteral="true">PublicAdministration</argument>
   </condition>
 </recipient>
 <recipient>
   <condition name="equal">
     <argument isLiteral="false">type</argument>
     <argument isLiteral="true">Non-ProfitOrg</argument>
   </condition>
 </recipient>
</recipients>
```

   The `recipients` element groups different `recipient` that are composed in OR logic among them, that is, an external party has to satisfy at least one of the `recipient` elements.

Purposes

The term *purposes* is used to denote those purposes for which the information can be used. Abstractions can be defined within the domain of purposes, so as to refer to purposes showing common characteristics and to a whole group with a name. Abstractions can therefore correspond to generalization/specialization relationships. For instance, `pure research` and `applied research` can be seen as specializations of `research`.

PII abstraction

Data types can be introduced as abstractions of PII to let data handling policies be expressed in terms of data types, rather than single properties of the user only. Data types can be organized hierarchically. For instance, in Figure 11.2, `cc_info` can be seen as an abstraction for the credit card information, which can include the `number`, `circuit`, and `expiration` attributes.
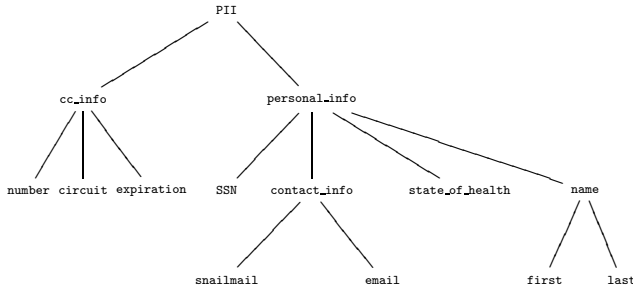
Fig. 11.2 An example of PII abstraction

Restrictions

A privacy statement specifies restrictions that have to be satisfied before access to personal data is granted. If just one condition is not satisfied, the access should not be granted. We distinguish between the following types of conditions.

- *Generic* conditions either evaluate properties of recipients' profiles, like membership of requester, or represent conditions that can be brought to satisfaction at run-time when the request is processed.
- *Provision* are preconditions that need to be evaluated as pre-requisites before a decision can be taken [BJWW02].

Syntactically, generic conditions and provision are boolean expressions of terms having the form `predicate_name(`*arguments*`)`, where *arguments* is a list, possibly empty, of arguments on which predicate `predicate_name` is evaluated. For instance, `in_area(`**user**,'New York'), is a generic predicate requiring **user** to be located within the metropolitan area of New York; `fill_in_form(`*form*`)` and `log_access()` are provision predicates that require to fill in a form and to log the access, respectively.

Each condition type (i.e., generic condition and provision) is defined by means of an ad-hoc XML element (see Appendix 30.2): `gen_conditions` and `provisions`. In the following we provide an example of generic conditions restricting the access from *10 am* to *2 pm*, and an example of provision stating that before access is given, it must be logged.

```
<gen_conditions>
 <condition name="time">
   <argument isLiteral="true">10am</argument>
   <argument isLiteral="true">2pm</argument>
 </condition>
</gen_conditions>
```

```
<provisions>
 <condition name="log_access">
   <argument/>
 </condition>
</provisions>
```

Obligations

Obligations are defined as complex policies inside data handling policies. They represent actions that have to be performed either after an access has been granted [BJWW02] or in the future based on the occurrence of well-defined events [Cas04b, CB07a] (e.g., time-based or context-based events). For instance, an obligation can state that users will be notified whenever their personal information is disclosed. Another obligation can impose a restriction on how long personal data should be retained (data retention). A full discussion of obligation policies is provided in Section 12.

11.5.1.1 Policy and Rule Definition

Syntactically, a data handling policy has the form "$\langle PII \rangle$ MANAGEDBY $\langle DHP\_rules \rangle$", where: *PII* identifies a PII abstraction that represents the name of an attribute or a data type, in case of a data handling policy template, a set of pairs of the form $\langle attribute\_name, attribute\_value \rangle$ belonging to a privacy profile, in case of a customized data handling policy; and *DHP_rules* identifies one or more rules, composed in OR logic, governing the use of PII to which they refer. Syntactically, a *DHP_rules* can be formalized as follow.

Definition 5 (Data handling rule). *A DHP_rules is an expression of the form $\langle recipients \rangle$ CAN $\langle actions \rangle$ FOR $\langle purposes \rangle$ $\big[$ IF $\langle gen\_conditions \rangle \big]$ $\big[$ PROVIDED $\langle prov \rangle \big]$ $\big[$ FOLLOW $\langle unique\_obl\_id \rangle \big]$, where:*

- recipients *can be an identifier, a category, or a boolean formula of* `credential` *and/or* `declaration` *predicates;*
- actions *is the set of actions;*
- purposes *is the purpose or a group thereof;*
- provisions *and* generic *conditions are optional boolean expressions of terms having the form* `predicate_name(`arguments`)`, *where* arguments *is a list, possibly empty, of arguments on which* `predicate_name` *is evaluated;*
- obligations *which are referred inside a data handling policy through a* unique_obl_id.

A data handling rule specifies that *recipients* can execute *actions* on *PII* for *purposes* provided that *prov* and *gen_conditions* are satisfied, and with obligations *obl*.

Table 11.2 An example of data handling policies that protect `Alice`'s data

Data Handling Policies		
PII	**DHP Rules**	**Description**
DHP1 `Alice.cc_info`	⌈credential(employeeCard(equal(**user**.job, 'Secretary'),equal(**user**.jobLevel,'A')), K_H) AND declaration(equal(**user**.company,'Hospital'))⌉ CAN read FOR service_release IF time(8:30am,6:00pm) PROVIDED log_access()	Secretaries of the hospital whole level is A can read credit card information of `Alice` for service release purpose during the working hours (i.e., from 8:30 am to 6:00 pm) provided that the access is logged.
DHP2 `Alice. personal_info`	⌈(declaration(equal(**user**.type, 'BusinessPartners')) AND declaration(equal(**user**.country,'EU'))) OR (declaration(equal(**user**.type,'GovAuth')))⌉ CAN read FOR research FOLLOW obl-id-001	European business partners of the hospital or government authorities can read `personal_info` of `Alice` for research with obligation obl-id-001.
	⌈credential(identity-document(equal(**user**.name.given,'John'), equal(**user**.name.last,'Doe'),K_H) AND declaration(equal(**user**.job,'Doctor')))⌉ CAN read FOR service_release IF in_area(**user**.sim,'Hospital')	Doctor John Doe can read the personal information of `Alice` for service release purpose only if he is in the hospital area.

Example 4. An example of data handling policies is provided in the simplified form shown in Table 11.2. Suppose that a `Hospital` provides services to its patients. Table 11.2 shows an example of customized data handling policies that regulate the secondary use of personal information of `Alice` stored by the `Hospital`. In particular, *DHP1* is composed of a single rule that protects the `cc_info` of `Alice`; *DHP2* is composed of two rules that protect the `personal_info` of `Alice`.

11.6 Related Work

A number of research works about privacy and identity management have been presented in the last few years. The lines of research closely related to the work in this chapter are in the areas of credential-based access control models, trust negotiation solutions, and privacy-aware models and languages.

Access control models based on digital credentials make decisions about whether or not a requesting party may execute an access on the basis of properties that this party may have. These properties can be proven by presenting one or more certificates [BS02a, NLW05, YWS03]. The first proposals that investigate the application of credential-based access control to regulate access to a server are done by Winslett et al. [SWW97, WCJS97]. Access

control rules are expressed in a logic language, and rules applicable to a service access can be communicated by the server to clients. A first attempt to provide a uniform framework for attribute-based access control specification and enforcement is presented by Bonatti and Samarati [BS02a]. The framework includes an access control model and a language for expressing access control and release policies, and a policy-filtering mechanism to identify the relevant policies for a negotiation. Access rules are specified as logical rules, with some predicates explicitly identified. Also, this proposal permits to reason about certified attributes, modeled as credential expressions, and declared attributes (i.e., unsigned statements). Communication of requisites to be satisfied by the requester is based on a filtering and renaming process applied to server policies, which exploits partial evaluation techniques in logic programs. Other works (e.g., [GPSS05]) have also investigated solutions for providing authentication and access control based on biometric systems and information [GLM+04]. In this context, Cimato et al. [CGP+08] propose a privacy-aware biometric authentication technique that uses multiple biometric traits.

Besides solutions for uniform frameworks supporting credential-based access control policies, different automated trust negotiation proposals have been developed [SWY01, YW03, YWS01]. Trust is established gradually by disclosing credentials and requests for credentials [GNO+04]. In [RZN+05, WSJ00, YW03, YWS03], the authors investigate trust negotiation issues and strategies that a party can apply to select those credentials to submit to the opponent party during a negotiation. Trust-management systems (e.g., Keynote [BFIK98], PolicyMaker [BFL96], REFEREE [CFL+97], and DL [LGF00]) use credentials to describe specific delegation of trusts among keys and to bind public keys to authorizations. They therefore depart from the traditional separation between authentication and authorization by granting authorizations directly to keys (bypassing identities).

In the last few years, as the need of privacy increases, a number of useful *privacy enhancing technologies* (PETs) have been developed for dealing with privacy issues. In this context, access control solutions enriched with the ability of supporting privacy requirements have been provided and some privacy-aware models and languages have been defined. The first objective of such solutions is to build an infrastructure that, on one side, regulates and restricts access to data, and, on the other side, allows users to protect their privacy by keeping a level of control over their data after their release to third parties. In this context, important issues to be considered concern the definition, management, and enforcement of privacy obligations. While the management of obligations can be a reasonably easy task when the events that trigger them are well defined and simple to capture, it becomes more complex in the case of privacy obligations triggered by the occurrence of events and conditions non-necessarily related to time or known transactions.

Relevant work has been done by W3C with its Platform for Privacy Preferences Project (P3P) [Cra02, Wor02]. P3P addresses the need of a user to

assess that the privacy practices adopted by a service provider comply with her privacy requirements. P3P provides an XML-based language and a mechanism for ensuring that users can be informed about privacy policies of the server before the release of personal information. Users specify their privacy preferences through a policy language, called A P3P Preference Exchange Language 1.0 (APPEL) [W3C02], and enforce privacy protection by means of a user agent, which compares and verifies whether the P3P policy conforms to user privacy preferences. P3P is important to shape (aspects of) the trust that people might have on the enterprise by verifying which privacy requirements they can fulfill. However, P3P is mainly a "front-end" mechanism, in the context of Web Services. In its current form it is "passive", that is, it only checks if people expectations are matched against promises made by the enterprise. It does not address the problem of allowing users to express fine grained privacy policies and obligations; nor provide mechanisms to deal with the execution and fulfillment of these privacy policies and obligations, and related constraints by enterprises. Last but not least, it does not define an enterprise framework for dealing with privacy policies.

Focusing on the problem of privacy management for enterprises, the Enterprise Privacy Architecture (EPA) [KSW02b] encompasses a policy management system, a privacy enforcement system, and an audit console. EPA is aimed at improving trust in enterprises e-business and provides a new approach to privacy that tries to help organizations in understanding how privacy impacts business processes. Specifically, the work in [SA02] introduces additional architectural details about EPA along with an interpretation of the concept of privacy obligations. This concept is framed in the context of privacy rules (policies) defined for authorization purposes. This approach is further refined and described in the Enterprise Privacy Authorization Language (EPAL) specification [AHK+03, AHKS02]. In general, EPAL consists of an XML-based markup language and an architecture aimed at formalizing, defining, and enforcing enterprise-internal privacy policies. It addresses the problem on the server side and supports the need of a company to specify access control policies, with reference to attributes/properties of the requester, which protect private information of its users. The current EPAL specification does not provide a format (or description) for obligations; obligations are purely a placeholder in the policy rule.

XACML by OASIS [eXt05] proposes an XML-based language to express and interchange access control policies. XACML is designed to express authorization policies in XML against objects that are themselves identified in XML. Also, XACML specifies the syntax and format of obligations, which by definition are included in the access control policies. In addition to the language, XACML defines both an architecture for the evaluation of policies and a communication protocol for message exchange.

Despite the benefits of all these works, none provides a complete solution for protecting the privacy of users and regulating the use of personal information in secondary applications. Our work tries to fill in this gap and provides

an access control infrastructure that supports users acting in distributed environments in the protection of their privacy and in the management of their information when released to external parties.

11.7   Conclusions

The definition of a privacy-aware access control system that regulates access to data/services still preserving the privacy of the involved parties is an important research direction and a practical pressing need. Existing proposals and traditional access control systems focus on the server-side needs of securing access to their resources. As a consequence, access control models and languages turn out to be very limited from a privacy point of view. We have defined an access control model and language for restricting access to resources/data managed by a service provider and release of PII managed by the users, which take advantage by integration with credentials, ontologies, and context information. Afterwards, we have defined a data handling model and language allowing users to pose restrictions on the secondary use of their private data when they are released to external parties.

In the next chapters, we will focus on related obligation policies and on assurance control policies that capture users' preferences and ensure that users' constraints and expectations (as well as constraints dictated by laws and legislation) can be explicitly represented in a language, and automatically enforced and checked by organizations. Furthermore, we will describe a prototype providing functionalities for integrating access control and data handling policy evaluation and enforcement together with a solution for obligation management and enforcement, and a solution for privacy compliance checking.

12

Privacy Models and Languages: Obligation Policies

Marco Casassa Mont

HP Labs

12.1 Introduction to Privacy Obligation Policies

Privacy obligation policies define and describe the expected behaviours and constraints to be satisfied by data receiving entities (e.g. enterprises, service providers, e-commerce sites, etc.) when handling confidential and personal data. In this section we will often refer to data receiving entities as enterprises. They dictate a privacy-aware identity lifecycle management including data retention and deletion aspects, management of notifications and requests for authorization, data processing and transformation workflows.

Enterprises need to put in place underlying IT infrastructures, processes and mechanisms to be compliant with these obligations. This can be a challenging task due to the fact that privacy obligations can differ quite substantially given their current level of refinement (abstract vs. refined) and their "multidimensional" nature involving multiple factors and aspects.

Privacy obligations can be very abstract and generic, for example: "every financial institution has an affirmative and continuing obligation to respect customer privacy and protect the security and confidentiality of customer information" – Gramm-Leach-Bliley Act [Act03].

This type of obligations dictates high-level principles and guidelines that need to be interpreted, refined and grounded to specific contexts in order to be fully understood in terms of their operational implications. More refined privacy obligations can be expressed in terms of:

- notice requirements;
- opt-in/opt-out options, limitations on reuse of information and information sharing for marketing purposes;
- data retention and deletion limitations.

At the other extreme, privacy obligations can dictate very specific requirements. This is the case where data retention has to be enforced for a long period of time or data are temporarily stored by organisations: privacy obligations can require that personal data must be deleted after a predefined number of years, e.g. 30 years (i.e. long-term commitment) – or in a few days if user's consent is not granted (i.e. short-term commitment). Other very specific privacy obligations might require the enterprise to notify (for example via e-mail) the data subjects, in case their data has been accessed by third parties or unauthorised people (for example in case of hacking or identity frauds). Similarly, privacy obligations might mandate to execute well-defined workflows and processes, involving both humans (e.g. for explicit request for authorization) and computer systems in presence of specific events.

12.2   Analysis of Privacy Obligations

Privacy obligations depend on and are influenced by a variety of aspects, including data subjects' preferences, enterprise guidelines, legislation and, once refined, technical aspects. Figure 12.1 is an attempt to capture this multi-dimensional nature of privacy obligations, based on our current analysis of privacy obligations [Cas04b, Cas04a] and their implications in terms of lifecycle management of personal data.
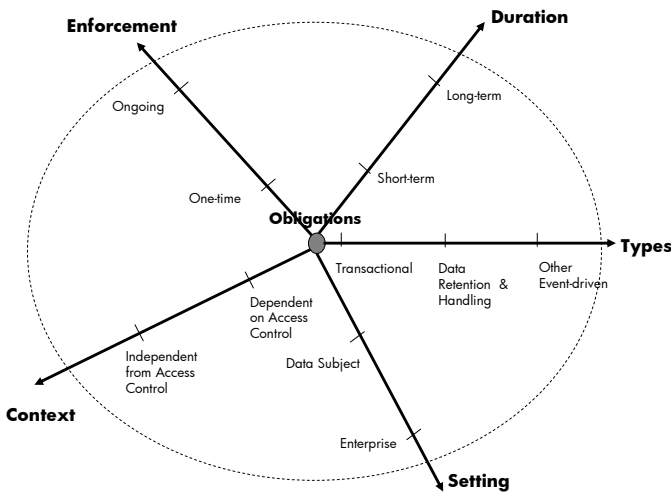
Fig. 12.1 A Multidimensional View of Privacy Obligations

The key aspects that need to be considered to characterise privacy obligations are:

- **Types of obligations:** obligations can be classified based on the fact that they are:
  - Transactional: these obligations need to be fulfilled immediately, during a transaction or interaction, when accesses to personal data are required;
  - Data Retention and Handling: these obligations are related to the management of personal data in terms of their deletion or transformation. They can be long-termed and unrelated to accesses to data;
  - Other Event-driven obligations: these obligations are triggered by events that can be dictated by contextual and system information, such as location of systems, their trustworthiness, aggregated meta-information associated with data (such a access counters, etc.);
- **Duration:** obligations can be classified based on their "lifetime", i.e. the period of time where they are active and subject to enforcement:
  - Short-termed: privacy obligations could be short-termed. This is the case of transactional obligations or obligations the lifetime of which ranges in the order of few hours to a few months;
  - Long-termed: privacy obligations could be long-termed. This applies to all cases where the data retention period could span to the order of years and consequently obligations need to be fulfilled over that period of time;
- **Enforcement:** obligations can be classified based on their enforcement implications:
  - One-time: this is the case where a privacy obligation can be considered as being fulfilled once it has been enforced. For example, an obligation dictating the deletion of a piece of data at a specified point in time belongs to this category;
  - Ongoing: this is the case where a privacy obligation might require to be "enforced" multiple times, during its lifetime. For example, this is the case of obligations dictating periodic notifications, over a predefined period of time;
- **Context:** obligations can be classified based on the context where they operate and are likely to be triggered for fulfilment:
  - Access control context: privacy obligations can be triggered as an effect of accessing data. This is the case, for example, for transactional obligations;
  - Access control-independent context: privacy obligations can be triggered in a context completely independent from access control, for example deletion of data at a due period of time;
- **Setting:** obligations can be set by different entities:
  - Data subjects: data subjects could define privacy obligations to be fulfilled on their data, for example by specifying opt-in, opt-out options

that are transformed into obligations for enterprises. This can include deletion and notification preferences. Alternatively, trusted third parties, acting on behalf of data subjects, could do this, for example identity providers in federated identity management contexts;

– Enterprise: administrators within the enterprise might define privacy obligations on data, as dictated by internal guidelines and/or legislation.

Figure 12.2 shows two simple examples of privacy obligations and their mapping into this multi-dimensional space.
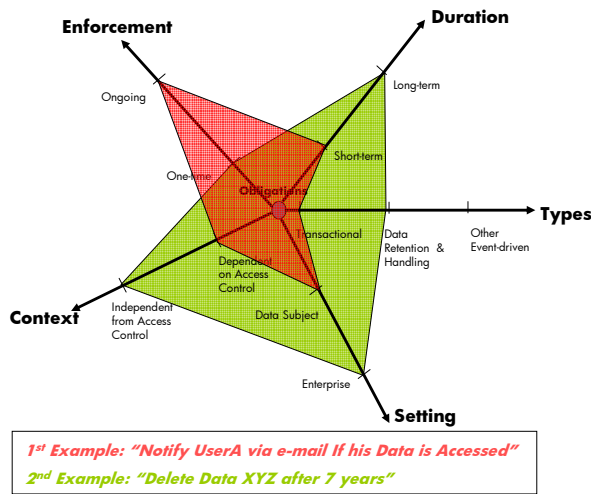
Fig. 12.2 Simple Examples of Privacy Obligations

The first example of privacy obligation, "Notify UserA via e-mail if his/her Data is Accessed", dictates data handling criteria. It can be set by the data subject on his/her account (for the entire lifetime of this account). It requires multiple enforcements (every time personal data is accessed). This obligation is triggered by accesses to the personal data. The second example of privacy obligation, "Delete Data XYZ after 7 years", can be set by an enterprise privacy administrator. It has long-term implications but it requires one-time enforcement (deletion of data at a predefined period of time). It is independent of access control aspects: data has to be deleted independent of the fact whether it has ever been accessed.

Our analysis of privacy obligations [Cas04b, Cas04a] is based on current privacy laws, privacy guidelines and customers' requirements. It has identified a set of core properties that are shared by privacy obligations:

1. **Period of validity of an obligation:** it is the lifetime of an obligation, i.e. the period of time where the obligation is "active" and needs to be managed (enforced and monitored);
2. **Degree of enforceability of an obligation:** the enforcement of privacy obligations can be automated or, in some cases, it might need to involve human processes and best practices;
3. **Target (involved data) of an obligation:** privacy obligations refer to personal data subject to these obligations. Different, heterogeneous types of data, stored in multiple data repositories, can be referenced by a privacy obligation;
4. **Events that trigger the need to fulfil an obligation:** privacy obligations can be triggered by one or more events (for example time-based events). Logical combinations of events (involving AND, OR and NOT operators) might be required to express the conditions under which privacy obligations need to be enforced;
5. **Actions that need to be executed to enforce an obligation:** the enforcement of an obligation might require the execution of one or more actions. These actions could be as simple as deleting data or notifying people or require the execution of complex workflows that might involve human and computer interactions;
6. **Entities that are responsible for enforcing an obligation:** for each obligation it should be clear who (organisation, group, individual) is responsible for their management and enforcement;
7. **Accountability criteria:** these criteria mainly define logging and auditing requirements, to ensure that the system keeps an historical track of how an obligation is managed and enforced and which violations occurred;
8. **Exceptions:** exceptional cases might need to be analysed and explicitly described, in order to assure a correct management and enforcement of obligations.

Part of these privacy obligations can be enforced by software systems, i.e. tools can be built in order to manage and automate their fulfilment, based on the expressed constraints and requirements. Other privacy obligations, dictating expected human behaviours, still need to rely on best practices and good behaviours of enterprises and employees. Nevertheless, we believe that the process of moving towards automation (for those obligations where this is possible) is useful to enterprises to help them in their governance, regulatory compliance and cost reduction efforts. In our work we focus on automatically-enforceable privacy obligations. Concepts and approaches described in the remaining part of this work can still apply to other types of privacy obligations, at least with regard to the modelling aspect and the analysis of related requirements. We specifically focus on the requirements and issues related to

the management and enforcement of the following three core categories of privacy obligations:

1. **Long-term privacy obligations**;
2. **Short-term and transactional privacy obligations**;
3. **Ongoing privacy obligations**.

Figure 12.3 shows a few examples of events and actions related to these types of privacy obligations.

Long-term Privacy Obligations				
Events Triggering Obligations			**Actions Dictated by Obligations**	
Time-driven	1.at a specific date and time (e.g. 1:00am 01-Jan-2005) 2.after a certain period of time (e.g. 1 hour, 3 days, 5 minutes) 3.after the data has being used for a certain number of times (e.g. after being used twice) in a specific timeframe		Delete/ Update	1.delete all confidential data of a given data subject 2.partially delete data (e.g. delete only the credit card number) 3.replace data with an updated set of data (e.g. update subject's address)
Driven by Usage and Counters			Hide/ Unhide	•hide (encrypt) all data of a subject from any access •hide a part of this data from any access •unhide all data •unhide a part of the data
Ongoing Privacy Obligations				
Events Triggering Obligations			**Actions Dictated by Obligations**	
Time-driven	1.periodically (e.g. every month)			1.send a report to a subject containing the status of their data and their opt-in/opt-out options (e.g. number of times being used, who has tried to access) 2.tell the subject what data he/she has provided 3.get updated data from subject 4.audit the logs, report any improper use of the data
Driven by Contextual Events	1.when the data being used 2.when the data being transferred 3.when the data being deleted 4.a particular party/parties try to access 5.data is being used for certain purpose (e.g. send advertisement) 6.a set of data is going to be retrieved together 7.any action predefined by the data subject		Notify	•notify the subject
			Log	1.take logs
			Access	1.default allow/disallow all access 2.allow 3.disallow
			Consult	1.get authorization from data subject 2.get authorization from third party 3.check according to certain condition made by the user
Others	1.when the privacy policies changed			1.Stop access to the data 2.update obligation
Short-term and Transactional Privacy Obligations				
Obligations might need to be dictated by a transaction or an interaction. The actions specified by these obligations might need to be immediately fulfilled. These actions can be the same as the ones specified by long-term and on-going obligations.				

Fig. 12.3 Types of privacy obligations and examples of related events and actions

12.3 Requirements and Constraints

To categorize core issues and requirements related to the management and enforcement of privacy obligations we analysed a few scenarios involving the management of digital identities and identified a few common patterns:

- **Enterprise scenario:** personal data are collected from customers, employees and business partners. They are accessed, used and processed to enable business transactions and processes. Data can be disclosed to business partners and/or third parties;

- **E-commerce scenario:** personal data are collected from customers, mainly to enable business transactions and for marketing purposes. Data can be disclosed to third parties;
- **Healthcare scenario:** medical and personal data are collected from patients. Data can be accessed by medical people and shared with third parties for research and medical reasons;
- **Government scenario:** personal and financial data are collected from citizens by government offices (Revenue Office, Pension Office, Home Security Office, etc.) to provide government services and for security reasons;
- **Federated identity management scenario:** this scenario is complementary and orthogonal to the above scenarios. It is about dealing with explicit federated environments, where personal data and identities are shared among multiple parties (usually within a circle of trust or based on contractual agreements) to enable single-sign-on and speed-up the authentication process.

In these scenarios data subjects (people) directly or indirectly disclose their personal data to enterprises (organisations). In doing so, they might be asked (or want) to specify their privacy preferences, for example in terms of opt-in/opt-out choices, requests for notifications, retention, usage and disclosure of their data for predefined purposes. Enterprises using modern identity management solutions can provide self-registration and user provisioning tools that allow users to retain control of part of their data and specify (and change over time) some of their requirements and preferences. Some of these preferences must be translated into explicit privacy obligations to allow for their automated management within organisations, such as obligations to notify data subjects about usages of their data, delete data, protect data, etc. A few important questions arise.

- How can privacy preferences be translated into privacy obligations?
- Which format should be used to represent privacy obligations?
- How are links and associations between privacy obligations and stored data going to be handled?

Privacy administrators within these enterprises might need to set up additional privacy obligations on stored data, to fulfil privacy laws and/or internal guidelines. This might apply to all information involving personal data, including data subjects' records, audit logs, documents, etc.

- Which tools are required by administrators to manage and check these obligations on a large database containing personal data?
- How would these tools fit in current identity management solutions?
- How to ensure that enterprises will handle these data and related obligations in an accountable way?

In all these scenarios, personal data might be exchanged across boundaries, e.g. with other organisations, to enable interactions, transactions or business

processes. If these data are subject to privacy obligations, obligations need to be communicated as well. In some cases they must be modified and adapted, depending on the location and nature of the data recipients.

How to ensure that privacy obligations are "strongly" associated with these data and will be enforced?

Our investigation identified the following important issues and requirements which need to be considered when dealing with the management and enforcement of privacy obligations:

1. **Explicit modeling of privacy obligations:** to be managed, privacy obligations need to be represented with an appropriate language to describe which data is affected by an obligation, the events and conditions that trigger the fulfilment of the obligation, actions to be carried on, which entities are responsible and accountable for their enforcement;
2. **Association of obligations to data:** the association of privacy obligations to the targeted confidential data must not be easy to be broken. This aspect is particularly challenging in dynamic environments where confidential data can be moved around or sent to other parties;
3. **Mapping obligations into actions:** when possible, actions and sequences of actions dictated by obligations must be expressed in a way that can be programmatically enforced; otherwise, they should trigger related processes and workflows involving the human intervention and clearly stated responsibilities;
4. **Compliance of refined obligations to high-level policies:** the mapping of high-level policies to refined privacy obligations (and the affected data) should be managed explicitly and tools built to spot potential inconsistencies and dependencies;
5. **Tracking the evolutions of obligation policies:** obligation policies can be carried on over long periods of time and are subject to changes. Changes need to be tracked and obligations versioned, for accountability reasons and to deal with the evolution of the contexts and frameworks where these obligations apply;
6. **Dealing with long-term obligation aspects:** long-term obligations have implications on the longevity and survivability of related processes and the involved data. Solutions need to be build to last over a long period of time;
7. **Accountability management:** as anticipated before, accountability management is fundamental to ensure that the enforcement of privacy obligations is carried on with clear responsibilities of the involved parties. This introduces requirements in terms of auditing, tracking of obligations and their monitoring;
8. **Monitoring obligations:** the fulfilment of obligations must be monitored and checked against expected situations and behaviours. Despite good intents and enforcement mechanisms, it can always happen that the fulfilment of obligations is omitted. Monitoring mechanisms must be

orthogonal to the enforcement mechanisms. Problems need to be notified to the responsible entities;

9. **User involvement and awareness:** users should have visibility of which obligations an organisation has with them. Tools should be provided to users to allow them to monitor their fulfilment and directly manage their privacy obligations;

10. **Complexity and cost of instrumenting applications and services:** the enforcement and monitoring of obligation policies can have an impact on the involved applications and services, both in terms of their instrumentation and development costs. A privacy obligation framework should reduce this impact to the minimum.

11. **Integration with current identity management solutions:** systems that manage and enforce privacy obligations must integrate with current state-of-the-art identity management solutions.

12.4 Model of Privacy Obligations

Based on the available requirements and the analysis of the limitations of current solutions, we introduce and describe an alternative privacy obligation management model where privacy obligations are considered as "first class" entities and introduce an explicit privacy obligation management framework to handle these obligations. The details about our model and related concepts follow.

An obligation management framework is introduced to explicitly handle privacy obligations. Figure 12.4 shows the conceptual model underpinning this framework.

In our model privacy obligations are independent entities that are explicitly modeled and managed to enable a privacy-aware lifecycle management of personal data. They are not subordinated to access control aspects. Data subjects can define privacy obligations and associate them to their personal data at the disclosure time (e.g. during a self-registration process) or at any subsequent time. Enterprise privacy administrators can also associate additional privacy obligations, for example dictated by laws or internal guidelines. In our model, the obligation management framework handles these obligations and their associations to personal data by providing the following core functionalities:

- **Explicit modeling and representation of privacy obligations:** a language/format is defined to explicitly represent privacy obligations in order to analyse them and reason about their implications;
- **Scheduling the enforcement of privacy obligations:** the system schedules which obligations need to be fulfilled and under which circumstances (events);

Fig. 12.4 Proposed privacy obligation management model

- **Enforcing privacy obligations:** the system enforces privacy obligations once they are triggered. The enforcement ranges from the execution of simple actions to complex workflows involving human interventions;
- **Monitoring the fulfilment of privacy obligations:** the system monitors and audits the enforced obligations, at least for a predefined period of time, to ensure that the desired status of data is not violated and to report anomalies;
- **Administration and lifecycle management of privacy obligations**.

These functionalities can be accessed by enterprise privacy administrators and potentially by data subjects, for example to monitor their personal data and check for privacy compliance.

Our model of privacy obligations can be analysed by means of different (but equivalent) views/perspectives:

- **Conceptual view**;
- **Formal view**;
- **Operational view**.

12.4.1   Conceptual View

From a conceptual perspective, a privacy obligation can be considered as an entity (object) with a few associated properties, as shown in Figure 12.5. In this view, a privacy obligation is characterised by the following core properties:

- **Obligation Identifier:** it is an identifier to uniquely identify an obligation within the entire obligation management system;
- **Targeted Personal Data:** it is a list of references to personal data that are affected by this privacy obligation. A reference must include all the information necessary to reach the data, though it can be codified in a way to avoid any indirect exposure (or correlation) of personal data.
- **Triggering Events:** it is a list of logical (AND/OR) expressions based on combinations of basic events (e.g. time, access, counters) that can trigger the need to enforce the privacy obligation;
- **Actions:** it is a list of actions to be executed at the enforcement time of the privacy obligation. Actions could be very simple – such as deletion of data or sending a notification – or much more complex, for example workflows involving both system and human interaction steps.
- **Additional Metadata:** it is a placeholder for additional properties still under exploration, such as exceptions, accountability constraints, versioning and integrity check, etc.
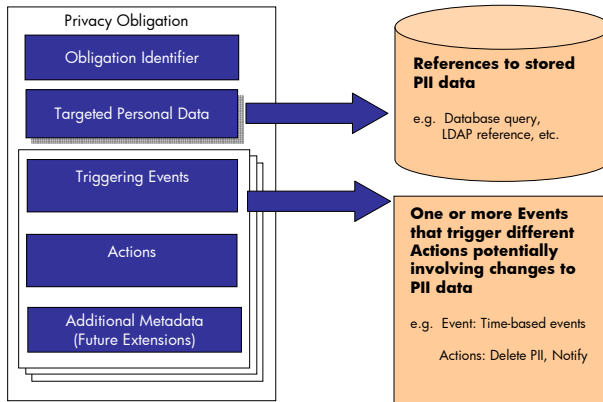
Fig. 12.5 Model of a privacy obligation

12.4.2   Formal View

From a formal perspective, a privacy obligation can be seen as a tuple $\langle i, t, L(e), C(a)\rangle$, where $\langle i, t, e, a\rangle \in \langle I, 2^T, 2^E, 2^A\rangle$:

- I: set of unique identifiers, associated to obligations;
- T: set of possible obligation targets, i.e. data entities (e.g. personal data, digital identities, attributes, etc.) subject to obligations;
- E: set of possible events that can trigger an obligation;
- A: set of all possible actions that can be executed as an effect of enforcing an obligation.

Specifically, a $\langle i, t, e, a \rangle$-tuple is defined as follow:

- $i \in I$: i is an element that belongs to I;
- $t \in T$: t is a set of targets included in T;
- $e \in E$: e is a set of events included in E;
- $a \in A$: a is a set of actions included in A.

A privacy obligation is obtained by applying the L-operator to the set e and the C operator to the set a:

- $L(e)$: defines a logical combination of events, for example AND, OR and NOT combination of events contained in e;
- $C(a)$: defines an operational combination of actions, such as a sequence of actions.

It is beyond the scope of this section to provide a systematic definition or formalization of privacy obligations. In this section we will have a pragmatic view of privacy obligations, based on how we can represent them and how we can operate on them.

12.4.3 Operational View

From an operational perspective, privacy obligations can be seen as reactive rules [RHCMP05], i.e. rules that are triggered by events and/or by the fact that the specified conditions are met. As an effect (reaction) of triggering a rule, actions are executed.

A representation of privacy obligations as reactive rules follows:

```
OBLIGATION Oid:
   TARGETS: t
   WHEN L(e)
   EXECUTE C(a)
```

In this context, given an obligation with unique identifier Oid and a target t, if the logical combination of events $L(e)$ is true, i.e. it triggers the rule, then the combination of actions $C(a)$ has to be executed. The remaining part of this chapter will focus on this operational definition of privacy obligations.

As anticipated at the beginning of this section, privacy obligations are associated with personal data and can be defined by data subjects and privacy administrators. A few simple examples of privacy obligations follow:

```
OBLIGATION Oid1:
TARGETS:
  t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc>
  WHEN (current_time= date1)
  EXECUTE <DELETE t1>
```

In this example, a customer record, stored in a specified table of a database, must be deleted at a well-defined point in time. This is a simple example of a data deletion obligation.

```
OBLIGATION Oid2:
TARGETS:
  t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc,
  ATTRIBUTES=(e-mail) >
WHEN (Access_Data_Event AND Access_Data_Event.data = t1)
EXECUTE  <NOTIFY BY t1.e-mail>
```

In this example, when an event (for example issued by an access control system) indicates that a specific customer's record has been accessed, a notification has to be sent to the customer, by using his/her e-mail address.

```
OBLIGATION Oid3:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc
ATTRIBUTES=(creditcard,e-mail)>
WHEN
     (current_time>date1)
         AND
     (NOT (Access_Data_Event AND Access_Data_Event.data = t1 ))
EXECUTE
    <NOTIFY BY t1.e-mail>
    <DELETE t1.creditcard>
```

In this example, if customer's data is not accessed after a predefined amount of time, an attribute (credit card) has to be deleted and the customer must be notified.

```
OBLIGATION Oid4:
TARGETS:
  t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc
  ATTRIBUTES=(creditcard,e-mail)>
WHEN
    (current_time>date1)
        OR
        ( (Access_Data_Event AND Access_Data_Event.data = t1 )
            AND
          (Access_Counter>n))
EXECUTE
   <DELETE t1.creditcard>
   <RUN WORKFLOW deprovision_user(t1.KeyValue)>
```

In this example customer's data is deleted and the customer account is de-provisioned (accounts deleted, access rights revoked, etc.) from various IT systems either at a specific point in time or after customer's data has been accessed more than n times.

```
OBLIGATION Oid5:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc,
 ATTRIBUTES=(e-mail)>
WHEN
  (current_time < date1)
    AND
  (time_counter > time_interval)
EXECUTE
 <NOTIFY BY t1.email>
 <RESET time_counter>
```

In this example, periodic (ongoing) notifications are sent by e-mails to customers, for example to notify them about the fact that the enterprise is retaining their personal data. This is an example of an ongoing obligation. Specific types of privacy obligations can be set-up by enterprise privacy administrators to handle personal data based on internal guidelines and/or laws. These privacy obligations can be triggered by internal events determined by contextual and infrastructural changes. A few examples follow.

```
OBLIGATION Oid6:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers>
WHEN
     (Event-intrusion_detected)
EXECUTE
   <ENCRYPT  t1>
   <NOTIFY admin>
```

In this example, we assume that an intrusion detection system is able to send alerts to subscribers (including our obligation management system) when intrusion attempts are detected. A privacy obligation can be triggered to protect the entire content of a "confidential" table by encrypting its content and notifying the administrator. This action can be seen as a best-effort, temporary solution to prevent that personal data are accessed by the intruder.

```
OBLIGATION Oid7:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers>
WHEN
     (Event-system_distrusted)
      AND
      (DATABASE.host =system_distrusted.host)
EXECUTE
```

```
<ENCRYPT  t1>
<NOTIFY admin>
```

Similarly to the previous obligation, this obligation is triggered by contextual changes. In this case, one of the systems hosting the database is classified as "distrusted" (because of a virus infection, locally detected spyware, installation of dubious software, etc.) by enterprise monitoring systems. If this event is sent to the obligation management system, this system can trigger the above obligation that will encrypt the data and notify the administrator. Again, this action can be seen as a best-effort, temporary solution to prevent that personal data is compromised.

```
 OBLIGATION Oid8:
 TARGETS:
    t1:< FILE=../audit_log, ATTRIBUTES=(TimeStamp, UserIPaddress, UserName)>
WHEN
  (time_counter > time_interval)
EXECUTE
 <ENCRYPT  t1.UserIpAddress>
<DELETE t1.UserName WHERE t1.TimeStamp<= current_time - 6 months>
<RESET time_counter>
```

This privacy obligation is defined by a privacy administrator to "purge" the content of an audit log file (for example created by a web server) of specific personal data, as dictated by internal guidelines (for example after six months) and encrypt another portion of the data that can be decrypted later on, in case of need. This is an example of ongoing obligation that is periodically triggered based on a predefined interval of time (for example every week).

All the actions described in the above examples of privacy obligations can (conceptually) be generalised as workflows. A workflow consists of one or more actions/tasks to be executed, in a specified order. In the remaining part of this thesis the concept of workflow is implied whenever "actions" of privacy obligations are discussed. Please notice that the language used in the above examples to describe privacy obligations is purely illustrative.

12.4.4   Relationships with AC/DHP Policies

As anticipated in Section 11 on the access control model and language, obligation policies can be seen as an aspect (or component) of data handling policies. In this context, an obligation defines constraints and conditions on how to handle personal data.

Most of these constraints need to be explicitly described in a language, to ensure they can be enforced. Becase of the nature of obligation policies (e.g. dictating data retention, data deletion and notification criteria), additional policy enforcement points are required.

The following section describes the policy language used in PRIME to represent obligation policies. Chapter 15 on privacy-aware identity lifecycle

management will then provide additional details about how the obligation policies can be enforced and managed.

12.5 Privacy Obligation Policies: Language

Privacy obligations are represented by using an XML format [W3C03], even if alternative formats are currently under exploration (including [W3C04]). For the time being, the XML-based format has been chosen as it is suitable for future extensions of the content of privacy obligations, in a modular way. At the moment the following categories of privacy obligations have been implemented:

- Transactional obligations;
- Short- and long-term obligations;
- Ongoing obligations.

The events that are currently supported are:

- Time-based events;
- Counter-based events;
- Access control-based events for well-defined pieces of personal data;
- AND/OR combination of the above events: the AND/OR operators apply to the logical evaluation of events (the fact that they happened means they are TRUE, otherwise they are FALSE) and/or constraints on events. For example, a constraint on a time-based event such as "current-time >Date1" is TRUE if the current time is greater than "Date1" and FALSE otherwise. In this work, for brevity, we will refer to "constraints on events" as "events".

The actions that are currently supported are:

- Deletion of data;
- Notification via e-mail;
- Triggering of workflow-based actions (external to the obligation management system);
- Sequences of the above actions.

The .dtd definition of the XML-based format used to represent the above types of obligations follows:

```
<!ATTLIST obligation
    oid CDATA #REQUIRED
>
<!ELEMENT obligation (target, metadata, events, actions)>
<!-- target -->
<!ELEMENT target (database)>
<!ELEMENT database (dbname, tname, data)>
<!ELEMENT dbname (#PCDATA)>
<!ELEMENT tname (#PCDATA)>
```

```
<!ATTLIST data
    attr (all | part) #REQUIRED
>
<!ELEMENT data (item*)>
<!ELEMENT item (#PCDATA)>
<!-- metadata definition -->
<!ELEMENT metadata (type, description)>
<!ELEMENT type ANY>
<!ELEMENT description ANY>
<!-- events definition -->
<!ATTLIST events
    operator (OR | AND | NOT) #REQUIRED
>
<!ELEMENT events (event*, events*)>
<!ATTLIST event
    id CDATA #REQUIRED
>
<!ELEMENT event (type, date?, item?, times?, period?)>
<!ATTLIST date
    now (yes | no) #REQUIRED
>
<!ELEMENT date (year, month, day, hour, minute, second)?>
<!ELEMENT period (year?, month?, day?, hour?, minute?, second?)>
<!ELEMENT times ANY>
<!ELEMENT year ANY>
<!ELEMENT month ANY>
<!ELEMENT day ANY>
<!ELEMENT hour ANY>
<!ELEMENT minute ANY>
<!ELEMENT second ANY>
<!-- actions definition -->
<!ELEMENT actions (action*)>
<!ATTLIST action
    id CDATA #REQUIRED
>
<!ELEMENT action (type, data?, method?, to?)>
<!ELEMENT method ANY>
<!ELEMENT to ANY>
```

A simple XML-based privacy obligation is shown below, based on previous examples described in this chapter:

```
<?xml version="1.0"?>
<obligation oid="43459345908605678">
 <target>
  <database>
     <dbname>oms_demo-customerdb</dbname>
     <tname>customers</tname>
       <data attr="part">
```

```
            <item>@key:UserId:uid123|att:creditcard</item>
            <item>@key:UserId:uid123|att:email</item>
            <item>@key:UserId:uid123|att:name</item>
    </database>
  </target>
    <metadata>
        <type>LONGTERM</type>
        <description>
        Delete creditcard AND Notify User
        WHEN current time = 2006:04:19 13:28:00
    </description>
    </metadata>
  <events>
    <event id="e1">
        <type>TIMEOUT</type>
            <date now="no">
            <year>2006</year>
            <month>04</month>
            <day>19</day>
            <hour>13</hour>
            <minute>28</minute>
            <second>00</second>
            </date>
    </event>
  </events>
  <actions>
      <action id="a1">
        <type>DELETE</type>
        <data attr="part">
          <item>creditcard</item>
          <item>name</item>
        </data>
    </action>
      <action id="a2">
        <type>NOTIFY</type>
        <method>EMAIL</method>
        <to>email</to>
      </action>
    </actions>
  </obligation>
```

The content of this privacy obligation is self-explicative. It is about a
privacy obligation that targets three fields in a database (i.e. creditcard, name,
e-mail) within a database record (associated to a customer), identified by a
record key (UserId field, uid123). It is a "long-term" obligation, requiring the
deletion of the creditcard and name fields at a predefined date and sending a
notification to the user via e-mail.

A slightly more complex example of an XML-based privacy obligation
follows:

```xml
<?xml version="1.0"?>
<obligation oid="57856745880978">
  <target>
    <database>
      <dbname>oms_demo-customerdb</dbname>
      <tname>customers</tname>
      <data attr="part">
        <item>@key:UserId:uid123|att:creditcard</item>
        <item>@key:UserId:uid123|att:email</item>
        <item>@key:UserId:uid123|att:name</item>
        <item>@key:UserId:uid123|att:address</item>
      </data>
    </database>
  </target>
  <metadata>
    <type>LONGTERM</type>
     <description>
            Delete creditcard AND Notify user
              WHEN
                 creditcard has been accessed 2 times
              OR
                 Either current time is 2006:04:19 13:28:00
                   OR
                 Address has been deleted
    </description>
  </metadata>
  <events operator="AND">
    <event id="e1">
      <type>ACCESS</type>
      <item>@key:UserId:uid123|att:creditcard </item>
      <times>2</times>
    </event>
    <events operator="OR">
       <event id="e2">
        <type>TIMEOUT</type>
        <date now="no">
          <year>2006</year>
          <month>04</month>
          <day>19</day>
          <hour>13</hour>
          <minute>28</minute>
          <second>00</second>
        </date>
      </event>
      <event id="e3">
      <type>DELETE</type>
        <item>@key:UserId:uid123|att:address</item>
      </event>
</events>
```

```
    </events>
    <actions>
      <action id="a1">
       <type>DELETE</type>
       <data attr="part">
         <item>@key: UserId:uid123|att:creditcard</item>
       </data>
      </action>
     <action id="a2">
       <type>NOTIFY</type>
       <method>EMAIL</method>
       <to>@key: UserId:uid123|att:email</to>
     </action>
    </actions>
</obligation>
```

This privacy obligation requires the deletion of the creditcard attribute and the notification of the data subject when one of the composite events happens. This obligation can be triggered when the credit card has been accessed twice or either the data subject's address has been deleted (hence it does not make anymore sense keeping information about the credit card, assuming that acquired goods must be physically delivered) or a specific point in time has been reached.

An example of ongoing XML-based privacy obligations is shown below:

```
<?xml version="1.0"?>
<obligation oid="476567765676452456">
  <target>
    <database>
      <dbname>customerdb</dbname>
      <tname>customers</tname>
      <data attr="part">
        <item>@key:UserId:uid123|att:creditcard</item>
        <item>@key: UserId:uid123|att:email</item>
        <item>@key: UserId:uid123|att:*</item>
      </data>
    </database>
  </target>
  <metadata>
    <type>ONGOING</type>
    <description>
       Periodically Notify User
         Every 30 days
           OR
         Every time creditcard has been accessed twice
    </description>
</metadata>
    <events operator="OR">
    <event id="e1">
```

```
        <type>OGPERIOD</type>
        <period>
          <days>30</days>
        </period>
    </event>
    <event id="e2">
        <type>OGACCESS</type>
        <item>creditcard</item>
        <times>2</times>
    </event>
  </events>
  <actions>
    <action id="a1">
      <type>NOTIFY</type>
      <method>EMAIL</method>
      <to>email</to>
    </action>
  </actions>
</obligation>
```

All the above privacy obligations can be programmatically interpreted and automatically handled by our obligation management system. The current XML-based syntax of privacy obligations is simple enough to be directly edited and understood by people. However, graphical tools can be built to automatically generate obligations in the required format, driven by inputs and preferences provided by users.

Based on our XML representation of obligation policies, we also proposed an obligation management framework model and a related obligation management system to interpret, schedule, enforce and monitor these policies. Our obligation management technology and framework was designed to allow users (at the time of disclosing their personal data or afterwards) to express privacy preferences (e.g. on deletion time of some of their attributes or notification preference) on how their personal data should be handled by the enterprise. Our obligation management system was then able to automatically derive and instantiate related obligation policies based on these privacy preferences. We achieved this capability by introducing the concept of *obligation policy templates*. In our approach, a template consisted basically of an obligation policy which contained simple "placeholders" in its Events and Actions sections [Cas04c]. Templates were defined upfront, by privacy administrators, to cover all the types of obligations supported by an enterprise. In this context, a template was instantiated just by replacing its placeholders with the actual privacy preference values (for example a deletion date or a notification preference, etc.). In this context an "instantiated" obligation policy was (1) uniquely associated to a piece of data and (2) it embedded privacy preferences in its Events and Actions sections. The resulting "instantiated" obligation policies were then scheduled, enforced and monitored by our obligation management system [Cas04c].

12.6   Parametric Obligation Policies

The implementation of our prototype (and a related demonstrator), related tests and feedback received by third parties helped us to identify another key problem: the scalability of this approach to obligation policies. On the one hand our approach provided great flexibility in defining a broad range of privacy obligation policies, potentially customisable to users' needs and directly associated to personal data. On the other hand for each piece of managed data (and related privacy preferences), one or more "instances" of our obligation policies had to be created and associated to this data, as shown in Figure 12.6.
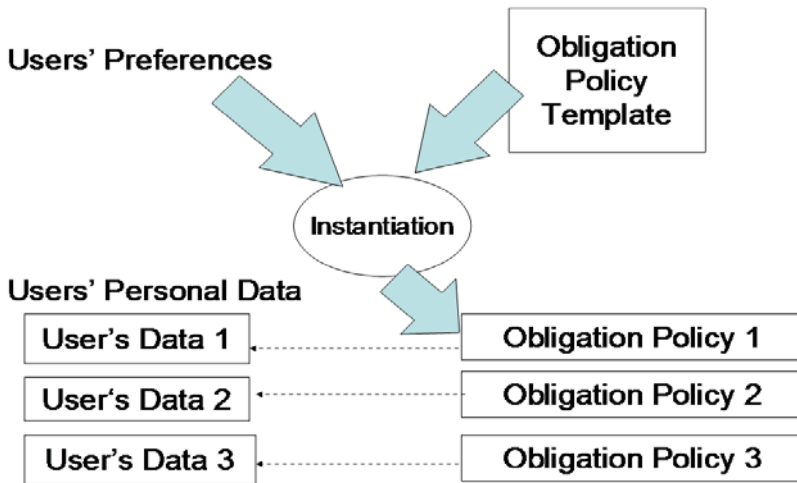
Fig. 12.6 Association of obligation policies with data

In real-world scenarios, large amounts of user data (greater than 100K records) are collected and managed by enterprises. In our approach, this meant having to deal with a similar (large) amount of associated obligation policies with negative implications and impacts in terms of required resources and processing power to run our obligation management system.

Additional feedback highlighted the need not only to passively monitor failures in enforcing privacy obligations (i.e. spotting cases where the enforcement of stated Actions fails or changes in the status of managed data invalidates previously enforced actions [Cas04c, Cas06]) but also being able

to proactively remediate to these failures (e.g. by notifying administrators or trying to reinforce failed actions).

Usability tests carried out on our obligation management system (by the Karlstad University) highlighted that end users are looking for simple ways to express their privacy preferences, via graphical user interfaces, on a well-defined, small and clear set of stated obligation policies. This finding reinforced the validity of our approach based on using pre-defined templates for privacy obligation policies, as a way to reduce the "types" of obligation policies to be managed in our obligation management system [Cas06]. This aspect was actually taken into account and implemented in PRIME.

However, the usage of templates, on its own, does not solve the scalability problem: even if the enterprise could just define a reduced set of obligation templates, these templates have nevertheless to be instantiated for each piece of managed data – based on related privacy preferences. Hence the scalability problem was still there. A complete analysis of this issue and other related aspects can be found in [Cas06].

The key problem that had to be addressed was how to manage obligation policies in a scalable way, on a potentially large set of personal data stored in various enterprise data repositories. The following related requirements must be satisfied (based on customers' feedback, our analysis and lessons learnt):

- Limit the number of "instantiated" policies (and related management resources) independently on the amount of managed data and related privacy preferences;
- Preserve the key capability to "customize" the management of each individual piece of personal data, based on users' privacy preferences;
- Provide a more comprehensive automation of obligation policies, ensuring that obligations (once enforced) are not only passively monitored but also actions are taken to remediate/react to any violation. This to reduce the need for human intervention in case of large datasets.

Addressing this problem has implication on two key aspects: (1) how to represent obligation policies; (2) how to manage, enforce and monitor these policies. Our approach and solution to the problem, based on the concept of parametric obligation policies, is presented next.

12.6.1   Parametric Obligation Policies: Model

To address the stated problem and keep into account related requirements, we introduce the concept of parametric obligation policies. A parametric obligation policy is a policy that leverages the concepts of our previous version of obligation policies [Cas04b, Cas04a, Cas04c]. The same categories of obligation policies are managed. However, the key differences are:

- A parametric obligation policy can be associated to a potentially large set of personal data (i.e. no multiple instantiations) and, at the same time,

it can dictate customized obligation constraints (based on users' privacy preferences) on each data item;

- A parametric obligation policy does not embed privacy preferences in its Events and Actions sections (as instead happens in our previous version of obligation policies). Instead, this policy contains explicit references to these preferences, that are stored elsewhere – in data repositories;
- The Target sections of parametric obligation policies explicitly model and describe the data repositories that will contain preference values pointed by these references – in addition to repositories containing personal data;
- A new "On Violation" section has been introduced to explicitly automate the process of "remediation" of violated obligations – as described in the section on requirements.

The key feature introduced by parametric obligations is that privacy preferences are stored separately from parametric obligation policies: references are used to retrieve these preferences. This ensures that a parametric obligation policy can apply to a potentially large set of personal data – as defined in its Target element – and, at the same time, allows the "customization" of its Events and Actions based on references to external privacy preferences.

From a formal perspective a parametric obligation policy is a tuple as follows: $\langle i, t, L(e[r]), C(a[r]), C(va[r]) \rangle$, where $\langle i, t, e, a, va \rangle \in \langle I, 2^T, 2^E, 2^A, 2^V A \rangle$ and $r \in 2^R$:

- I: set of unique identifiers, associated to parametric obligation policies;
- T: set of possible obligation targets, i.e. data entities subject to obligations;
- E: set of possible parametric events that can trigger an obligation, i.e. events that might contain references (e.g. to privacy preferences);
- A: set of all possible parametric actions that can be executed as an effect of enforcing an obligation, i.e. actions that might contain references (e.g. to privacy preferences);
- VA: set of all possible parametric "on violation" actions to be executed to remediate any violation of enforced (parametric) obligations. These actions might contain references to preferences as well;
- R: set of all possible references (e.g. to privacy preferences) that could be used in a policy.

Specifically, this tuple $\langle i, t, e, a, va \rangle$ is defined as:

- $i \in I$: i is an element that belongs to I;
- $t \in T$: t is a set of targets included in T;
- $e[r] \in E : e[r]$ is a set of parametric events included in E;
- $a[r] \in A : a[r]$ is a set of parametric actions included in A;
- $va[r] \in VA$: $va[r]$ is a set of parametric "on violation" actions included in VA;
- $r \in R$: r is a set of references (to values) included in R.

In this context the *L*-operator and the *C*-operator mean the following:

- $L(e[r])$: a logical combination of parametric events, for example AND, OR and NOT combination of events contained in e;
- $C(a[r])$: a combination of parametric actions, such as a sequence of actions;
- $C(va[r])$: a combination of parametric actions to be executed in a sequence, when an enforced obligation is violated.

A set of parametric obligation policies can be created by a privacy administrator to dictate the "criteria" by which personal data should be handled: the referencing mechanism (coupled to appropriate data descriptions in the Target section) ensures that these policies are "instantiated" on-the-fly by our obligation management system – based on associated privacy preferences, enforced and monitored on a potentially large set of managed data.

12.6.2 Parametric Obligation Policies: Reference Scenario

We consider an enterprise scenario where a potentially large number of users (customers, employees, etc.) have to disclose their personal data in order to access services. This personal data is provided by users at registration time, potentially via a web-based self-registration service. In this context a user can check which obligation policies (e.g. in terms of deletion of data, data minimization, notifications, etc.) the enterprise can support (and on which data). The user can make decisions on opting-in/opting-out some of these obligation policies (others might be mandatory). For each selected obligation policy the user can instantiate specific privacy preferences and submit the overall information. The user could later on access this "registration" web service and make changes to their personal data, selected obligations and privacy preferences. A privacy administrator, in the enterprise, can set additional obligation policies (derived from laws and/or internal guidelines) on any subset of collected personal data. The enterprise can enforce these obligation policies on managed data, by means of a privacy-aware information lifecycle management solution that leverages our approach and technology. This automates the enforcement of these policies, their monitoring and remediation activities (in case of violation of policies).

12.6.3 Parametric Obligation Policies: Language

A parametric obligation policy is still represented in an XML format, as a reactive rule. XML has been used because of its versatility and suitability to extensions. The XML skeleton of a parametric obligation policy is presented next:

```
<?xml version="1.0"?>
<obligation oid="">
      <target>...</target>
```

```
        <metadata>...</metadata>
        <events>...</events>
        <actions>...</actions>
        <onViolation>...</onViolation>
</obligation>
```

The remaining part of this section provides more details about the actual content of parametric obligation policies i.e. their **Target, Metadata, Events, Actions, OnViolation** sections. For illustration purposes we consider a simplified scenario. This scenario consists of an e-commerce site that collects personal data about users and their preferences and stores this information in database tables. In this context, we consider a very simple parametric obligation policy dictating the following: for each piece of managed personal data (Target), credit card information must be deleted (Parametric Action) based on time-based deadlines specified by users via their privacy preference (Parametric Event). When this happens the correspondent user must be notified (Parametric Action). Should the enforcement of any of these actions fail, the obligation management system should try to reinforce them and notify an administrator ("On Violation" Actions).

The **Target** section of a parametric obligation policy is used to provide the following information:

- A description of data repositories containing (personal) data that is subject to privacy obligations. In this context one or more data repositories can be described (e.g. RDBMS database or LDAP directory, etc.). A data repository description includes location and name of the data repository, data schema structures (e.g. database tables) and primary keys. It is important to notice that, by default, all data stored in these repositories will be affected by this obligation policy. A more selective choice of which data items must be managed can be made by instantiating a "Conditions" sub-section (e.g. by testing properties/values of the stored data). Each data repository is identified by a unique alias that is used as a shortcut in other parts of the parametric obligation. If multiple data repositories are described, it is possible to specify any relationship (i.e. links between primary keys) existing on data stored in these repositories;
- A description of data repositories used to store privacy preferences. The definition of this sub-section is identical to the previous one, with the exception that it refers to repositories storing preferences/parameters. These preferences are associated to the managed personal data and used to customize other sections of the privacy obligations;
- A cross-links sub-section defining how to link preferences to personal data, by using relevant keys defined in the other two sub-sections.

The XML skeleton of the **Target** section (low-level details have been omitted for space reasons) follows:

```
<target>
    <DataRepositories>
        <Repositories>
            <DataRepository alias= "..."
                <DRType>...</DRType>
                <DBname>...</DBname>
                <TableName>...</TableName>
                <Conditions>
                        <Condition>...</Condition>
                 </Conditions>
                 <UniqueIdentifier>
                        <References>...</References>
    </UniqueIdentifier>
            </DataRepository>
            <InternalLinks>
                  <Link>...</Link>
            </InternalLinks>
     </DataRepositories>
    <PreferenceRepositories>
            <Repositories>...</Repositories>
            <InternalLinks>
                    <Link>...</Link>
            </InternalLinks>
    </PreferenceRepositories>
    <CrossLinks> </CrossLinks>
</target>
```

In case of our simple example of privacy obligation policy, the above skeleton could be instantiated with the following information: (1) a data repository entry, containing the database and table names where personal data is stored, the table's "primary key" (e.g. UserId) and an alias (e.g. DataRepAlias) for this repository; (2) a preference repository entry, containing the database and table names where preferences are stored, the table's "primary key" name (e.g. PrefId) and an alias for this repository (e.g. PrefRepAlias). A field in this table, for example called TimePreference, could be used to store users' preferences about deletion time of Credit Card details; (3) a description (in the "Cross link" sub-section) of how to link personal data to preferences (e.g. DataRepAlias.UserId = PrefRepAlias.PrefId)

The **Metadata** section of a parametric obligation policy describes: (1) Type of obligation policy (e.g. "Parametric"); (2) Natural language description of the obligation, presented to users and/or administrators. The XML skeleton of the metadata section follows:

```
<metadata>
    <type>Parametric</type>
    <description>...</description>
</metadata>
```

The **Events** section of a parametric obligation policy describes "parametric" events that must occur to trigger the obligation. These events can contain references to personal data and preferences described in the Target section. The high level XML skeleton of the Events section follows:

```
<events operator="AND/OR/NOT ">
      <event id="e1">
         <type>...</type>
         </event>
    </events>
```

One or more event or events sub-sections can be described in this section, in a recursive way, combined via logical AND/OR/NOT operators. Each "event" subsection has a unique, local identifier. The actual definition of these events depends on their types. Currently managed types of events are:

- Time based event: it describes a condition that checks the current time (NOW) against a stated time. The "stated time" can be retrieved via a reference (e.g. to a field in a Privacy Preferences data repository);
- Data Access event: it describes a condition on how many times a specified user's data item(s) can be accessed in a predefined period of time. The actual information (user's data item, number of accesses and period of time) can be retrieved via references to values stored somewhere;
- Data Deletion event: it describes a condition that is true when a specified piece of data has been deleted (by an external system). The location of this data can be specified via a reference.
- Context-based event: it describes conditions on contextual information (e.g. system attributes, OS- or application-based information). References to this information can be used.

In our example of privacy obligation policy, a simple time-based event is described as follows:

```
<events operator=" ">
   <event id="e1">
     <type>TIMEOUT</type>
           <date">
           NOW > [#ref] PrefRepAlias.TimePreference
       </date>
        </event>
    </events>
```

In our example, the "NOW >[#ref] PrefRepAlias. TimePreference" condition is verified if the current time (NOW) is greater then a time accessible via the "[#ref] PrefRepAlias.TimePreference" reference. This reference points to information stored in the Privacy Preferences repository (having the PrefRepAlias alias) in the "TimePreference" field, as declared in the Target (see the Target example). It is important to notice that, in our example, each piece

of data has an associated preference value – specified by the user and stored in the Preference Repository ("TimePreference" field). At this "declarative" stage, this reference is a "generic" reference to potentially many values stored in the Preference Repository. It must be contextualized to each specific "piece of data" the policy applies to. This happens at "runtime", during the interpretation of events. Our scalable obligation management system will achieve this by using the Target section of this policy: for each targeted piece of data it will retrieve the associated preferences based on the specified reference (e.g. "TimePreference" value in the Preference Repository) and check any related condition in the events section (in our simple example it is a simple time-based condition). This is done in an efficient way, via a few SQL queries to databases. In our example, when the time-based condition is satisfied for a given piece of data and an associated preference, the system triggers the enforcement of related actions (on that piece of data).

The **Actions** section of a parametric obligation policy describes "parametric" actions to be enforced when an obligation is triggered by its events. These actions can contain references to data and preferences consistently with the definitions in the Target section. A high-level XML skeleton of the Actions section follows:

```
<actions>
    <action id="a1">
        <type>...</type>
        <onCondition> </onCondition>
         ...
    </action>
</actions>
```

One or more action sub-sections might be defined in this section. Each "action" sub-section has a unique, local identifier. Actions are executed in a sequence, potentially subject to the satisfaction of (optional) conditions (e.g. constraints on Privacy Preferences. By default these conditions are TRUE, i.e. actions are just executed). The actual definition of these actions depends on their types. Currently managed types of actions are:

- Notification Action: this action sends a notification to an entity. The e-mail address of this entity can actually be a reference to a value in the Data Repository;
- Deletion Action: this action deletes a piece of personal data or some of its attributes. A reference can be used to identify this piece of data;
- Command Execution Action: this action executes an external application or service (e.g. a workflow application to process a piece of data or transform it). References to personal data or privacy preferences can be passed as parameters;
- Logging Action: this actions logs information (including referenced information) for auditing purposes.

In our example of privacy obligation policy, two actions are defined, to delete user's credit card details and notify users:

```
<actions>
     <action id="a1">
         <type>DELETE</type>
<data attr="part">
    <item>
          [#ref] DataRepAlias.CreditCardRef
       </item>
    <item>
           [#ref]DataRepAlias.CreditCardNumber
       </item>
 </data>
      </action>
  <action id="a2">
     <type>NOTIFY</type>
               <method>EMAIL</method>
     <to>  [#ref] DataRepAlias.Email </to>
               <text> some e-mail text here </text>
     </action>
</actions>
```

These actions contain references to personal data (credit card details and e-mail address). The same observations made in the "Events" section apply here. These references are "solved" at runtime, based on contextual information, i.e. specific pieces of personal data for which obligations have been triggered.

The **On Violation** section of a parametric obligation policy describes "parametric" actions to be executed in case an enforced policy is violated, i.e. if any of its enforced actions fail. The XML skeleton follows:

```
<onViolation>
     <ovAction id="ova1">
        <type>...</type>
        <onCondition>...</onCondition>
        ...
     </ovAction>
    </onViolation>
```

An action can fail either at the enforcement time or afterwards (e.g. deleted data could reappear because of wrong database synchronisation): this latter case is detected by the monitoring component of our obligation management system. All actions described in the "Actions" section can be used in the "OnViolation" section. A specific "RE-ENFORCE" action has been introduced just for the "OnViolation' section: when used, it requires the system to re-enforce just the actions that have failed (in the Actions section).

12.7 Discussion

In this section we provided an overview of the concepts underpinning obligation policies, related requirements and a language to represent these obligation, to enable their automatic enforcement and management.

Our approach and work has been pragmatic, driven by real-world requirements and needs. This approach helped us to identify that scalability issues had to be addressed and suggested ways to move towards a parametric approach for obligation policies.

Work done in PRIME (in terms of architectures and prototypes) demonstrated how privacy-aware access control and obligation policy management are complementary aspects and can be successfully combined to: (1) improve an organisation's privacy practice; (2) provide better control to end users; (3) ensure that users' constraints and preferences can be automatically captured, enforced and monitored over time.

12.8 Next Steps and Future R&D Work

Next steps include carrying on further R&D work work in the space of obligation management, both in terms of policy representation and obligation management systems that fully manage and enforce these policies.

A future objective is to move towards standardised representation of obligation policies. This is particularly important when obligation policies have to "stick" to (be associated with) personal data when this data is exchanged between entities, e.g. enterprises or federated services. In this case, a common language and format will ensure a simpler management and enforcement of these policies, along with a common understanding of related policy constraints and preferences.

Another objective is to ensure that this work on obligation policies can be commercially exploited. We are looking at producing commercial solutions that can be integrated with state-of-the-art enterprise Identity Management solutions.

13

Privacy Models and Languages: Assurance Checking Policies

Siani Pearson

HP Labs

Assurance policies express the security and data protection processes and mechanisms which should be in place to protect users' data. Users define such policies to express the minimum privacy protection which they wish to have in place by the recipient of their data. Service providers publish their policies to assert protection which should be in place, and for which they are able to provide evidence that this is indeed the case. Thus, assurance policies may be regarded as a specialised form of release and data handling policy, depending upon the context.

In this section we explain the motivation for using assurance policies, and show some formalisms used within PRIME.

13.1   Introduction

Assurance checking policies are formulated by people to obtain degrees of assurance from enterprises that their data will be processed according to their expectations, such as compliance to privacy, security and IT standards. In many cases, the user is specifying the type of device and environment where their PII data is being viewed. These would usually be checked up-front before PII was released, either in the preamble or in a negotiation phase before release of PII.

Assurance checking policies are separate from obligations and are constraints and conditions usually expressed by people before they engage with enterprises. They might just specify a particular regulatory context or, more generally, can require enterprises to provide degrees of proof about their ability to:

- Support the enforcement of predefined privacy policies and obligations with respect to laws and legislation

- Run their processes, services and data repositories in a secure way
- Use secure and trusted systems, such as trusted computing platforms, to increase the level of security and trust in their operational activities.

Assurance checking policies may also be expressed by the services side, to obtain degrees of assurance from third parties that any data shared with these third parties will be processed according to the service provider's expectations, and also to express how data provided to the service provider by users will be processed, and to provide evidence that this is actually the case.

In summary, the overall motivation for defining assurance policies is to allow people to make judgements about the trustworthiness and privacy compliance of the remote receiver of their PII data. For example, a user might check for the compliance of an organisation against customised preferences prior to disclosure of PII data. Their disclosure of PII data or the continuation of a business interaction could be subject to the outcome of this checking.

13.1.1   Principles

The principles underlying the use of assurance policies are as follows:

- The assessment goes beyond just promises on behalf of the service provider
- It assesses the levels of proof that can be provided about privacy-providing mechanisms that are used and even how they are operating
- A broader range of information and assurance is assessed than just security information
- In some cases assurance policies can be checked independently of access control
- Broadly speaking, the conditions checked are necessary, not sufficient for the transaction to proceed
- Ultimately, it is the user who decides how to proceed

13.1.2   Natural Language Examples

The following natural language expressions give examples of the type of constraints people may want to express using assurance policies:

- "the processing platforms must give a high level of protection for my PII data"
- "the back end must have a valid privacy seal issued by a provider I (or some authority I delegate) trusts"
- "the back end must give tamper-resistant protection to secrets"
- "the service provider should support obligation management"
- "my PII will only be processed within EU"

13.1.3   Overview of Different Potential Approaches

There are a range of different approaches to defining assurance policies. We shall consider two approaches, both of which we considered within PRIME:

1. Trust and assurance constraints can be represented as first order predicate logic expressions: for example, `hasValidPrivacySeal(Issuer) & isTrusted(Issuer)`. These would be conjoined to other conditions within the 'conditions element' of data handling policies, transfer policies, etc. In particular, we may extend the access control representation given in Section 11.4.3 by defining a set of trusted-based predicates of the form `predicate_name(`*arguments*`)`. In this case, the assurance control checking is invoked as part of the Access Control Decision assessment within PRIME (see Section 11.4.3). Here, the assurance policy can be injected into the existing access control policies as predicates. For example: `IF (Access Control checks) AND IF (Assurance Control checks)`
2. An alternative approach is to use a much higher-level representation in which individual human-readable clauses within the policies are first class objects, thus defining a completely separate policy representation language from the other approaches presented in this chapter. Here, the checking is invoked as an independent assurance control function invoked by an entity to conduct tests against a compliance template: for example, a user initiating compliance checking of a website against their "e-commerce" policy.

Accordingly, there are different possible levels of representation that can be used within assurance policies, respectively:

1. At a low level e.g. $\forall x \in$ `ProcessingSystem (hasWorkingTPM(x)` $\vee$ `...) & usesAdequateEncryption(x) & isPatched(x) & hasWorkingOMS(x)`, with reference to a lower level semantics
2. At a high level e.g. `checkTrustedProcessingSystem`, with reference to a higher level semantics

The result of the assurance checking is an analogous structure to the assurance policy input (with the resulting values of the assurance clauses). The complexity can be shown to the user if desired, and the structure can simplify to a Boolean value, for instance so that the access control decision point may make a decision about what to do next.

The following sections consider these different approaches in turn.

13.2   Defining Trust Constraints: A Lower Level Representation

Our initial PRIME implementations used the notion of trust constraints [Pea06]. Trust constraints are part of a broader representation of constraints
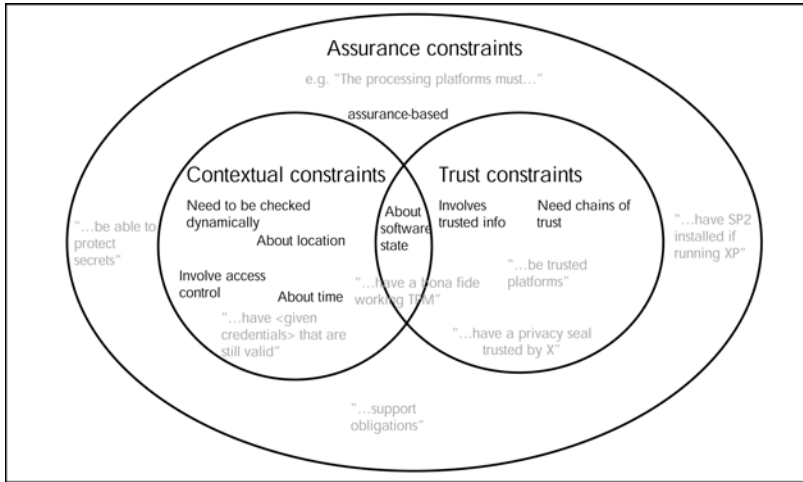
Fig. 13.1 Trust and assurance constraints

within policy languages. Figure 13.1 illustrates how, within the context of policies and preferences, they are a subset of a broader set of constraints about data processing: assurance constraints.

Assurance constraints may be contextual constraints, i.e. formulated by people to restrict the cases in which their data will be processed, according to parameters that may vary dynamically (such as time, location or platform state), or trust constraints. Figure 13.1 shows how these can be related and provides some examples. To a greater or lesser degree, all assurance constraints could be regarded as trust constraints since something must ultimately be trusted to make the assertions (and indeed the policy compliance checker must be trusted to issue compliance statements), but for convenience we may distinguish those statements that directly involve trust-related information and for whose automated evaluation a compliance checker needs to take chains of trust into account.

These constraints may be expressed within user-side preferences or policies. Such policies (*assurance policies*) would then include a set of conditions and constraints formulated to obtain degrees of assurance from enterprises that their data will be processed according to people's expectations, such as compliance to privacy, security and IT standards. On the client side the assurance control module (AC) can check their satisfaction (via information provided by the service-side AC) prior to disclosing any personal information or during the negotiation process (when AC provides input to both the local user and service-side requester). It can also be desirable to check contextual constraints on the service-side after information has been disclosed. This would be through the use of sticky policies, which are negotiated using the preferences and then associated with data as it travels around, perhaps just using a weak binding, although preferably this would use a strong binding

provided by cryptographic mechanisms: see for example [CPB03]. Furthermore, such constraints may be expressed within service-side access control policies to help enterprises comply with privacy legislation, such that service-side AC will not allow a transaction to be continued unless the constraints are fulfilled.

To clarify exactly what we mean by assurance policies (or constraints), let us consider the W3C Platform for Privacy Preferences (P3P) [Wor02] and Enterprise Privacy Authorisation Language (EPAL) [IBM04] schemas representation of privacy policy rules. These rules are formed of six elements, namely data user, data item, action, purpose, conditions and obligations. Assurance policies could be thought of as an extension to privacy policy rules in that they contain certain trust, contextual or assurance constraints which, if fulfilled, are not sufficient for the transaction to proceed: semantically, these constraints are necessary conditions. An example of an assurance policy which is an access control policy would be:

```
subject with subjexp can action on object with objexp if
condition onlyif assurance constraint.
```

(Alternatively, such an assurance policy could be represented by conjoining the assurance constraint to each subcondition.) There can be other, similar, forms of assurance policy, such as a policy that is attached to data and that contains assurance constraints that must be satisfied before certain actions may be performed on the data.

Within PRIME, we used this approach to define an assurance policy by defining trusted-based predicates (including those shown in Figure 13.1) within the access control policies (for further details see Section 11.4.3). Within the PRIME implementation, when such constraints were presented for evaluation to the access control module, the trusted-based predicate (i.e. the logical expression involving assurance constraints) would then be passed to the assurance control module for evaluation, and the result passed back to the access control module in order to calculate the overall policy satisfaction.

Assurance constraints (including trust constraints) can also be thought of in an orthogonal sense as breaking down into subconstraints, such that there can be functional decomposition of higher-level privacy and trust goals into one or more lower-level goals, and so on recursively until facts about the knowledge base (e.g. checks about the value of constraint settings, the presence of software, the availability of services for a given minimum uptime, etc.) are invoked at the lowest level. This decomposition is captured by rules within our system that hook into the PRIME ontologies used so that the meaning can be agreed across multiple parties. For example, even a fairly low-level trust constraint such as that the receiving party should use tamper-resistant hardware to store key information must be defined in such as way as to make clear the manufacturers, version numbers and other ancilliary information such as degree of tamper resistance that would or alternatively would not be acceptable. In practice, a third party would define such rules in advance and

then they would be viewable and/or customisable if desired at a later stage by users or administrators. See [Pea06] for further details about this approach.

13.3   Defining Clauses as First Class Objects: A Higher-Level Representation

The above representation was used within the initial phases of the PRIME project. However, in the final phases we refined this approach to replace it by making clauses be first class objects. We found this approach to be preferable to the previous approach, because humans need to read and interpret the assurance policies. Therefore, we wished to tilt the balance in favour of ease of understanding, with the clauses being expressed in natural language, at the expense of the expressivity and richness of the language. We wished to push the complexity of the checking to the third parties involved in the production of evidence, and make things as simple as possible for the end users.

Our model of assurance policies can be analysed by means of different (but equivalent) perspectives, namely the conceptual view, the formal view, and the operational view. We consider these views in turn, and provide examples of assurance polices and the language used.

13.3.1   Conceptual View

Both users and service providers have the freedom to create policies to suit their needs. In order to bring the two together a common vocabulary is developed. This comes in the form of privacy statements or privacy clauses which are a basic primitive of our solution. A clause is a statement concerning a particular privacy aspect of PII. It is succinct, clear, and unambiguous and clearly communicates its intended purpose at a level that does not require expert knowledge of privacy systems or their implementation. It is expressed in natural language with the aim that both clients and services will be able to understand each other more clearly. This empowers an end-user, of whom it is assumed to not have technically advanced knowledge, to communicate their privacy preferences in a language they understand. Later in this chapter we discuss how our policies relate to previous work on policy definition.

A policy is a collection of clauses, crafted for a particular purpose depending on the context of the interaction. Both the user and service provider will invoke the policy that they feel is the most appropriate depending on the context. For the user interacting with a bank they may invoke an "on-line banking" policy; for a service provider interacting with an on-line shopper they may invoke a "website customer" policy. The policies will be geared towards making sense of the context in which they are used. So an "on-line banking" policy may have stricter and more numerous clauses than a "signing up for free email account" policy. It is up to the user and service provider to maintain a pool of polices and invoke them under the proper circumstances.

This should then marry up the amount of processing and level of assurance required dependent upon the situation.

There is still an issue about where the clauses come from in the first place, and who provides guidance or establishes what is an appropriate policy for a particular purpose and what is not. In order to facilitate both problems it is important that there be some agreement about privacy in general and clauses and policies in particular. A way of doing this is through standardization. Trusted entities, such as governments or standardization bodies such as the W3C, who have experience in this field through efforts like P3P, can be called upon to provide a working pool of clauses and provide guidance on how to go about creating a privacy policy that is appropriate for a particular activity as a template (see further discussion below).

This approach has an important quality which we have dubbed *privacy positive*. A privacy positive statement is one that is privacy friendly. The clauses are created carefully and worded in such a way to be privacy positive: that is to say that a clause will never reduce the level of privacy afforded to the individual.

The predefinition of clauses is important for three reasons.

1. The clauses are not concerned with technical implementation details: only statements about privacy as required by law, good business practice, and consumer protection will be present. This abstracts away the technical details from the essence of the statements which are only concerned about what should happen with PII and not how it should happen. It prevents restrictions on the way the solutions are implemented and also prevents users from having to be technically savvy to use this scheme.
2. To protect users from having to understand technical details about privacy products and construct detailed policies which may be removed from practical reality, users use clauses that they care about to fashion their policies. In the same vein a service provider, although more technically knowledgeable, uses the same clauses and can speak the same language as its users and can communicate its responsibilities clearly.
3. Since the same pool of statements are being used by both the users and service providers it is an easy matter to match up expected policies with actual ones and negotiate the mismatches. At least in this way the glaring omissions in service providers' policies will become obvious and in the same way unrealistic expectations from users can be cleared up. Where there are deficiencies in specific clauses, the totality of the policy must be looked at. The set of clauses that form the policy is a stronger indication of the suitability of a policy than the individual clauses of which it is made up. Even if there is disagreement between a user and the service provider at least both know where the other stands on privacy.

We are aware that positive and negative clauses are subjective but it is hoped that through proactive efforts by lawyers and privacy experts in concert

with privacy groups it is possible to arrive at a standard of privacy expectations and conduct.

The idea of an *assurance policy template* is an optional extension to this approach, which can be convenient for users in order to help them build up their assurance policies with the help of entities that they trust. An assurance policy template can be thought of a set of default policies (or more specifically, clauses suggested to the user to include within their assurance policy), associated with a given context. For example, an assurance policy template might be suggested by a consumer group for a particular scenario (e.g. purchasing goods of value less than $100 online), and this template would list the checks that the consumer group recommended making in that case. There could be different templates for different contexts, and potentially more than one template for a given context; it would be up to the user to select the appropriate template which they wished to use, if any — this could be done automatically in fact, once the users' initial choices about which templates to use if different preconditions matched were selected and stored.

Templates for policies can provide a set of clauses that adhere to best practices or commonly held standards. To this a user can add or remove clauses depending on their preferences and needs. Templates are especially geared towards end users who may need help creating a privacy policy that would serve the purposes that the end user needed them for.

For example a template for on-line banking may recommend that:

1. PII remains confidential in transit
2. PII is only accessed by authorized personnel
3. A valid privacy seal is present
4. PII is not released to third parties without the consent of the user.

The template can be used in a policy editor to further ease the creation of policies. The end user could use the template as a solid starting point and then tweak it to their desires. This way they can concentrate on their privacy concerns rather than worry about technologies and get distracted from their original intentions. Similarly, a business could also use templates to the same effect although it would have to be careful to only include those clauses it had the actual capability to enforce. A business could not include a clause it could not honour into its policy since the involvement of trusted third parties (TTPs), to be discussed in Section 16.2.7, prevents this type of abuse.

13.3.2   Examples of Clauses

Examples of privacy positive clauses can be about any aspect of privacy, from:

- We will not share your data without your consent

to

- We will delete your PII after 30 days

A clause will not break common privacy expectations or allow circumvention by statements such as:

- We will share your data with third parties

or

- We reserve the right to store your data indefinitely

Such privacy negative clauses do not add to the privacy of consumers and would not be valid in privacy policies or adopted in the standard clause pool.

13.3.3   Formal View

From a formal perspective an assurance policy template can be seen as a $\langle ctid, pc, L(cc) \rangle$ tuple, where $\langle ctid, pc, cc \rangle \in \langle CTID, PC, CC \rangle$ and where $L(cc)$ defines a logical combination of cc, such that:

- $pc \in PC$: set of all preconditions
- $cc \in CC$: set of all assurance/compliance clauses
- $ctid \in CTID$: set of all unique identifiers

An assurance policy is a $\langle L(cc) \rangle$ list, where:

- $\langle cc \rangle \in \langle CC \rangle$
- $cc \in CC$ is the set of all assurance clauses

Further formalisation of this view is beyond the scope of this section.

13.3.4   Operational View

From an operational perspective, assurance policy templates (aka. compliance checking policy templates) can be seen as collections of compliance clauses. A representation would be:

```
CCPT ctid:
 IF  <pc>
Check L(cc)
```

In a similar way, assurance policies (aka. compliance checking policies) can be seen as collections of clauses as determined by pre-conditions:

```
CCP ctid:
   Check L(cc)
```

For example, *Transfer selected PII if it adheres to Government Policy Template* might correspond to:

```
CCPT ctid1:
IF  templateIS(Government)
CHECK
inEU(ReceivingLocality) AND trusted(ReceivingParty)
```

and *Transfer sensitive selected PII only if it adheres to Secure Storage Policy Template* might correspond to:

```
CCPT ctid2:
IF  templateIS(SecureStorage)
CHECK
NOT(isSensitive(t1)) OR (trustedPlatform(Device) AND
            encrypted(t1,minLevel))
```

13.3.5   Representation of Assurance Policies in XML Format

Within PRIME, we have used an XML format to represent assurance policies. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ccpolicy SYSTEM "ccp.dtd">
<ccpolicy>
<ctid>1337</ctid>
<clause gid="1">
<option>1285</option>
<option>AES</option>
</clause>
<clause gid="2">
<option>128</option>
</clause>
<clause gid="3">
</clause>
</ccpolicy>
```

where `ctid` is a unique identifier to identify each policy, `gid` is a unique well-known global identifier whose mapping to human-readable form is standardised, and `option` is a refinement to the clause if applicable.

The corresponding representation of (an example of) the results of the compliance checking process is:

```
<ccpolicy>
<ctid>1337</ctid>
<clause1 gid="1">
<constraint1>128</constraint1>
<constraint2>AES</constraint2>
<result>Y</result>
```

```
<signature>abcdef1234567</signature>
</clause1>
<clause2 gid="2">
<constraint1>128</constraint1>
<result>Y</result>
<signature>1341341234124</signature>
</clause2>
<clause3 gid="3">
<result>Y</result>
<signature>98765787646</signature>
</clause3>
</ccpolicy>
```

The DTD schema underpinning this XML format is:

```
<!ELEMENT ccpolicy (ctid, clause*) >
<!ELEMENT ctid (#PCDATA) >
<!ELEMENT clause (option*) >
<!ELEMENT option (#PCDATA) >
<!ATTLIST clause gid CDATA #REQUIRED >
```

The clauses in the standard clause pool are stored in any suitable form, e.g. a string, or a URI, and each of these is associated with a natural number that is the clause global identifier (`gid`) so that they can be referenced efficiently.

The user assurance policies specify which of these clauses within the standard clause pool the user wishes to check, and the service-side assurance policies specify which clauses within the standard clause pool the service provider is willing to testify that it can provide. These assurance policies are of the same form.

13.4   Analysis

During research it was found that assurance ontologies were not the best candidate for assessing the trustworthiness of the back end. The modelling of back end systems was difficult due to problems of classifying technologies and processes into a coherent assurance ontology that captured all types of systems and variations that are present in real world deployments. Also, this meant that there was a direct link between the technology in use by back-end systems and privacy policies, since privacy polices had to express privacy in terms that back end systems could understand. Another related problem was that even if the model were perfect and complete there existed an expectation that the end user be competent enough to gauge how these technologies benefited them. To avoid these two problems, standardized privacy clauses were introduced, the functionality of which was discussed above.

A policy in the context of this section is the formalization of another party's privacy compliance request. Here, the 'policy' parameter is taken very broadly,

and could just include references, or could be a very rich structure. There has been a great deal of work done on privacy polices [Wor02, HJW02, KSW02a, KSW03, MB03]. In these policy frameworks the focus has been on access control based on conditional logic. Our policies are a departure in that they are not processed against some rule set to produce a decision on whether data should be released. Rather, polices are just collections or groupings of clauses that serve a particular purpose under a particular context. Our solution takes into account that access control plays a big part in the control of PII and so the Assurance Control component works in concert with other components in the PRIME framework, namely Access Control Decision Function (ACDF) and Identity Control (IDCTRL), to address a variety of aspects needed within a privacy solution, from setting privacy preferences and handling PII requests, to controlling PII release.

P3P is a W3C specification that allows websites and end users to specify their privacy practices and preferences respectively in a standardized way that are easy to retrieve and interpret by end users. It allows a user to delegate the privacy policy "reading" by software agents that compare retrieved website polices against the one created by the user. Only policies that are in violation are flagged to the user who must decide what to do. There have been many critiques of P3P such as [Clab, Ack04, HJW02, Claa]. We shall ignore politico-economic arguments and focus on how our solution differs from P3P, the gaps it fills in, and how P3P could be used within the system we have implemented albeit with changes to its role.

Expressing privacy concerns in P3P is done by defining statements in a machine readable format written in XML [Wor02]. Although there are editors [P3P] that help with this process, there are two problems that are not yet addressed.

First, the P3P language and editor are tools but the end user must know what they wish to express in the first place. They must know what their privacy vulnerabilities are and how to check if a website will mitigate those risks. Most users are naïve and would not be competent enough to express privacy concerns beyond vague statements.

Second, even with the prerequisite privacy knowledge the definition of privacy polices must be in a language geared towards the facilitation of accessing PII based on conditions. Although useful, it cannot capture other aspects of privacy adequately without losing some of the essence of what the end user intended. Our solution addresses these concerns by introducing standardised privacy clauses that are written in human-readable form and are unambiguous, concise, and capture privacy concerns based on expert knowledge. To ease the creation of polices, templates are provided. The end user does not need to learn a language or an editor that requires knowledge of predicate logic.

As is the case with privacy seals, P3P cannot link the privacy practices expressed by the website with anything tangible on the back-end. This gap is where our solution introduces mechanisms to check that policies and the technical realities of the website's infrastructure are coherent. Claims made

in the privacy policies are backed up by capability checks as described in Section 16.2.5 and help to provide assurances that are missing from the P3P model.

Although P3P has its limitations, its strength as a robust policy definition language and logic model allows it to perfectly translate privacy clauses into machine-readable form. The resultant privacy policy would have to be vetted by TTPs and also certified, or an intermediate layer could be introduced that would drive the policy editor to receive clauses and output machine-readable policies. Since the clauses are defined and standardised the resultant XML would also be identical. In our model unique global identifiers are used to identify particular clauses, the drawback being that a unique identifier needs a lookup table to be maintained, whereas an XML policy would capture all the necessary information within itself. There have to be extensions to the present P3P vocabulary so that all aspects of privacy can be expressed.

13.5   Next Steps and Future R&D Work

A policy in the context of this section is the formalization of another party's privacy compliance request. Here, the 'policy' parameter is taken very broadly, and could just include references, or could be a very rich structure.

We have used a common standardized privacy clause pool to help communicate end user concerns as well as service provider promises. These clauses form high-level assurance checking policies. The benefits of this approach are in providing flexibility, and in being extensible and customisable. Chapter 16 gives details of the framework that maps these policies to back-end technology in such a way that this abstracts the complexity away for the end user and at the same time allows the service providers flexibility in how they implement and manage their infrastructure.

Since trust is not a black and white issue, we designed this approach such that the user must have overall control over how to proceed. Nevertheless, there is subjectivity and potential changeability of the decisions and representations involved, which could be an issue.

Since clauses are the central privacy vector they need to be developed further from the select set that are being implemented now. They need to be more complex and recognise complex privacy needs of sophisticated users as well as laws and regulations that businesses must adhere to. They also need to be stated in such a way that is unambiguous in any language. Only the true essence of the privacy objective of the clause must be present in its description. This will be an interesting area which will require participation from law, business, and security experts for further refining and establishing a coherent, effective, and simple language for defining privacy issues and concerns.

Further details about assurance control are given in Chapter 16.

Privacy-Aware Access Control System: Evaluation and Decision

Claudio Agostino Ardagna, Sabrina De Capitani di Vimercati,
Eros Pedrini, and Pierangela Samarati

Università degli Studi di Milano

14.1 Introduction

The success of the Web as a platform for the distribution of services and dissemination of information makes the protection of users' privacy a fundamental requirement. The privacy issues affect different aspects of today's Internet transactions, among which access control represents the most critical. An important step towards the protection of privacy is then the definition of a privacy-aware access control system that, in addition to server-side resources protection, provides users with solutions for preserving their privacy and managing their data. Although considerable work has been done in the field of access control for distributed services [AHK+03, AHKS02, BS02a, eXt05, Wor02], available access control mechanisms are at an early stage from a privacy protection point of view. This situation reflects the fact that in the last years the variety of security requirements focused on addressing server-side security concerns (e.g., communication confidentiality, unauthorized access to services, data integrity). Here, we focus on the development of a privacy-aware access control system regulating access to resources and protecting privacy of the users.

Generally speaking, an environment well-suited for users that need a private and secure way for using e-services should support at least the following basic requirements.

- *Privacy.* A digital identity solution should be respectful of the users' rights to privacy and should not disclose and manage personal information without explicit consent.
- *User-driven constraints.* In addition to traditional server-side access control rules, users should be able to specify constraints and restrictions about the usage that will be made of their information once released to external parties.
- *Minimal disclosure.* Service providers must require the least set of credentials needed for service provision, and users should be able to provide credentials selectively, according to the type of online services they wish to access.
- *Interactive enforcement.* A new way of enforcing the access control process should be defined based on a negotiation protocol aimed at establishing the least set of information that the requester has to disclose to access the desired service.
- *Anonymity support.* As a special but notable case of minimal disclosure, many services do not need to know the real identity of a user. Pseudonyms, multiple digital identities, and even anonymous accesses must be adopted when possible.
- *Legislation support.* Privacy-related legislation is becoming a powerful driver towards the adoption of digital identities. The exchange of identity data should not violate government legislations such as the Health Insurance Portability and Accountability Act (HIPAA) or Gramm-Leach-Bliley Act (GLB).

In the following, we present the prototype of a privacy-aware access control system, which supports the above requirements and integrates traditional access control mechanisms with release and data handling policies. In particular, we focus our discussion on policy evaluation and composition. Our privacy-aware access control system deals with five main key aspects: *i) resource representation,* ability to specify access control requirements about resources in terms of available *metadata* describing them; *ii) subject identity,* the evaluation of conditions on the subject requesting access to a resource often means accessing personal information. This raises a number of privacy issues, since electronic transactions (e.g., purchases) require release of a far greater quantity of information than their physical counterparts; *iii) secondary use,* although users provide personal information for use in one specific context, they often have no idea on how such personal information may be used subsequently. Users should be able to define restrictions on how their information will be used and processed by external parties; *iv) context representation,* context information is a set of metadata identifying and possibly describing entities of interest, such as subjects and objects, as well as any ambient parameter (including location) concerning the technological and cultural environment where a transaction takes place. As far as policy enforcement is concerned, context contains information enabling verification of policy conditions and,

therefore, it should be made available to any authorized service/application at any time and in a standard format. A major factor harnessing the potential of context representation is the lack of a standard context representation metadata layer; *v) ontology integration,* a privacy-aware access control should exploit the Semantic Web to allow the definition of access control rules based on generic assertions defined over concepts in the ontologies, which control metadata content and provide abstract subject domain concepts [ADD+05]. A central element of semantic-aware privacy policies is the use of semantic portfolio supporting controlled access to contextual resources (e.g., personal, company, and public services) subject to user-specified privacy constraints.

The remainder of this chapter is organized as follow. Section 14.2 presents the interactions between parties for data and services release. Section 14.3 describes the architecture of the access control module. Section 14.4 presents how the policies are evaluated. Section 14.5 describes the prototypes of *Access Control Decision Function* and *Policy Management* components. Section 14.6 analyzes the performance of the decision process.

14.2   Interplay between Parties

The infrastructure in Figure 11.1 is aimed at managing two different interplays between the involved parties (see Figure 14.1):

- *User-Service Provider interplay*, when a User submits an access request for a resource managed by the Service Provider, and
- *Service Provider-External Party interplay*, when an External Party submits an access request for sensitive information of a User stored by the Service Provider.

The access request submitted by a client or an external party can be defined as follows.

Definition 6 (Access request). *An* access request *is a 4-tuple of the form* ⟨user_id, action, object, purposes⟩, *where* user_id *is the optional identifier/ pseudonym of the requester,* action *is the action that is being requested,* object *is the object on which the requester wishes to perform the action, and* purposes *is the purpose or a group thereof for which the object is requested.*

For instance, the access request ⟨*Alice, execute, book_a_flight, service_access*⟩ states that *Alice* wants to *execute* the service *book_a_flight* for the purpose of accessing the requested service (*service_access*). On the other hand, the access request ⟨*Lufthansa, read, Alice_Credit_Card_Number, service_release*⟩ is a typical example of an External Party request, where external party *Lufthansa* wants to *read* personal data of a client.
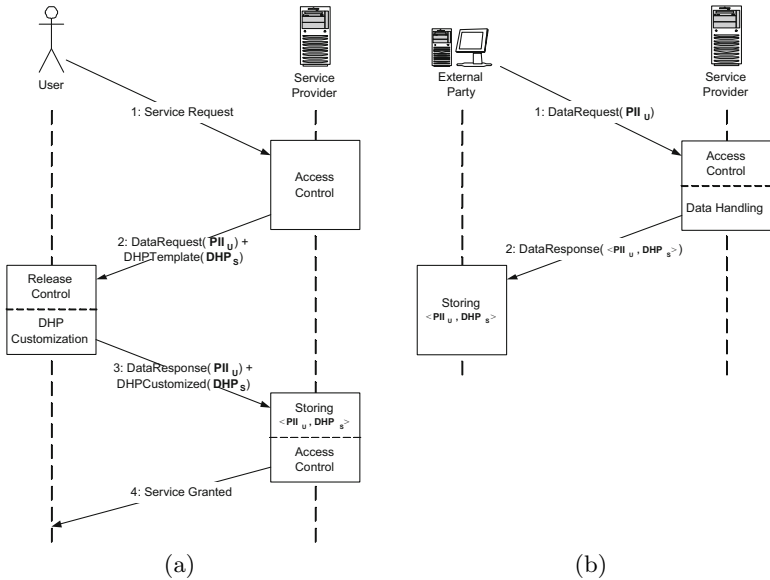
Fig. 14.1 User-Service Provider (a) and External Party-Service Provider
(b) interplays

User-Service Provider Interplay

The User-Service Provider interplay is depicted in Figure 14.1(a) (step 1-4).
Upon the reception of a service request (step 1), the service provider eval-
uates its access control policies and, if needed, requests some *PII* from the
user ($DataRequest(\text{PII}_U)$ in step 2); this happens if the information pro-
vided by the user is not sufficient for taking an access control decision. In
this case, the service provider presents a data handling policy template to the
user for customization ($DHPTemplate(\text{DHP}_S)$ in step 2). The user receives
the service provider request, evaluates her release policies, and customizes
the data handling policy template. If the *PII* request satisfies at least one
(user-side) release policy, the user sends back the required *PII* (PII_U) along
with the customized data handling policy (DHP_S) (step 3). Otherwise, if the
user's release policies deny the *PII* release, the transaction aborts. Finally,
the service provider stores the user's data $\langle\text{PII}_U, \text{DHP}_S\rangle$ and re-evaluates the
access control policies based on PII_U. If the evaluation succeeds, the service is
granted to the user (step 4). In a more general setup, both the release of *PII*
and the data handling policy customization could require multiple negotiation
steps [YWS01], rather than a single request-response exchange. The user, for
example, may require the service provider to release some *PII* as well, and,

in turn, the service provider may want to specify a data handling policy for such a *PII*.

Service Provider-External Party Interplay

Service Provider-External Party interplay (see Figure 14.1(b)) is temporally subsequent to the interplay between the user and the service provider. Suppose that an external party requests access to personal information of a user (i.e., PII_U) stored at the service provider ($DataRequest(\mathrm{PII}_U)$ in step 1). The External Party-Service Provider interplay begins with a mutual identification, followed, in the most general case, by a negotiation of the data to be released and attached data handling policies.[1] Differently from the previous interplay, in this case the service provider must protect the privacy of the user against the external party, by enforcing its access control policies and the data handling policies attached to the requested information. If the evaluation of access control and data handling policies returns a positive answer, the external party obtains the pair $\langle \mathrm{PII}_U, \mathrm{DHP}_S \rangle$ (step 2). Otherwise, the access is denied.

14.3   A Privacy-Aware Access Control Architecture

Figure 14.2 shows the privacy-aware access control architecture and related modules. Among them, the Access Control module is composed by two main components: *i) Access Control Enforcement Function* (*ACEF*) that is responsible for enforcing access control decisions by intercepting accesses to resources and granting them only if they are part of an operation for which a positive decision has been taken; and *ii) Access Control Decision Function* (*ACDF*) that is responsible for taking an access decision for all access requests directed to data/services.

In the remainder of this section, we focus our discussion on *ACDF* and *Policy Management* (i.e., the modules providing the privacy-aware functionalities), describing their characteristics and the interactions with others components.

14.3.1   Access Control Decision Function

The *Access Control Decision Function* (*ACDF*) component is responsible for taking an access decision for all access requests directed to data/services. *ACDF* produces the final response possibly combining the access decisions coming from the evaluation of different policies (access control, release, and data handling). The elements relevant to a decision are applicable *policies*,

---

[1] For the sake of clarity, Figure 14.1(b) does not show these steps that are similar to steps 2 and 3 of the User-Service Provider interplay in Figure 14.1(a).
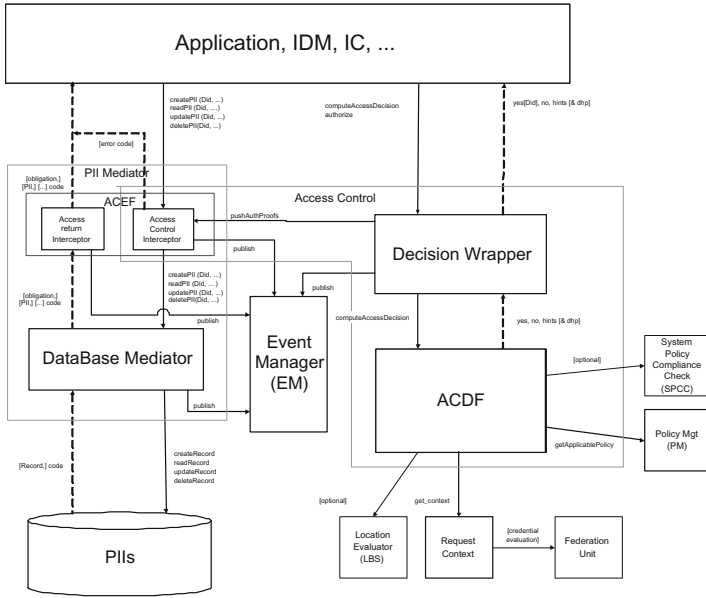
Fig. 14.2 Privacy-aware access control architecture

context which contains the information associated with the requester during a session, and *subject*, *action*, *object*, and *purpose* of the access request (i.e., the 4-tuple introduced in Definition 6). The decision component can return three different decisions:

- *Yes*: the request can be granted;
- *No*: the request must be denied;
- *Undefined*: current information is not sufficient to determine whether the request can be granted or denied. In this case, additional information is needed and the counterpart will be asked to provide such information.

ACDF mainly interacts with two modules: *Request Context* and *Policy Management* (*PM*). The *Request Context* module keeps track of all contextual information, aggregates information from various context sources, and deducts new contextual information from this aggregation. The main tasks of the *Request Context* module are then to provide contextual information from various sources in a standardized way, and to provide reasoning functionalities for boosting the evaluation process. Note that, *ACDF* does not interact directly with the *Request Context* module, but it relies on a Facade[2] component, called *Data Reader*. The Facade component has been designed to

---

[2] The Facade pattern [GHJV95] encapsulates a complex subsystem within a single interface object, and decouples the subsystem from its potential clients.

simplify the process of retrieving the information needed by *ACDF* for the evaluation. This solution adds a level of isolation that guarantees the simple integration of *ACDF* with different context formats or modules. The *Request Context* module also interacts with *Federation Unit*, a credential verification module in charge of verifying X.509 certificates and IDEMIX credentials [CL01c, CV02] (i.e., anonymous credentials) released by the counterparts. Only verified certificates are stored by the *Request Context* module, and used for policy evaluation.

The *Policy Management* module manages, stores, and distributes the policies to be used in the access control evaluation process. *ACDF* communicates directly with *Policy Management* for retrieving the policies applicable to an access request. We extensively discuss *Policy Management* in the next section.

ACDF also interacts with the *LBS Evaluator* and *System Policy Compliance Check* (*SPCC*) modules. *LBS Evaluator* is in charge of evaluating location-based conditions [ACD+06], such as `in_area(`**user**`.Sim, 'Milan')` that restricts access to requesters located in downtown Milan. The *SPCC* is in charge of handling assurance policies created by users and service providers. A policy in this context is the formalization of another party's privacy compliance request.

14.3.2   Policy Management

The *Policy Management* (*PM*) module manages the overall policy life cycle by providing functionalities for administering policies. Also, it is the module that interacts with the *Access Control* (*AC*) module for filtering responses coming from *AC*, and for restricting the release of sensitive information related to the policy itself or to the status against which the policy has been evaluated. The definition of sensitive information and sanitization operations are issues managed in cooperation with the *AC* module. The *Policy Management* module addresses the following requirements:

- it provides operations for policy administration, such as search, store, update, check, and delete;
- it provides functions for searching policies applicable to a certain request;
- it provides filtering functionalities that restrict the release of sensitive information related to the policies, when additional requests have to be generated from AC;
- it provides policy storage.

As shown in Figure 14.3, the *PM* module includes the *Policy Presentation* (*Ppres*) and the *Policy Processing* (*Pproc*) components. The *Policy Presentation* component acts as a policy presentation interface that receives as input an access request, and returns as output the applicable policies. This interface is used by *ACDF* to take an access decision. Note that *Ppres* communicates directly with *Policy Repository* to retrieve policies. The *Policy Processing* component provides filtering functionalities on the response that
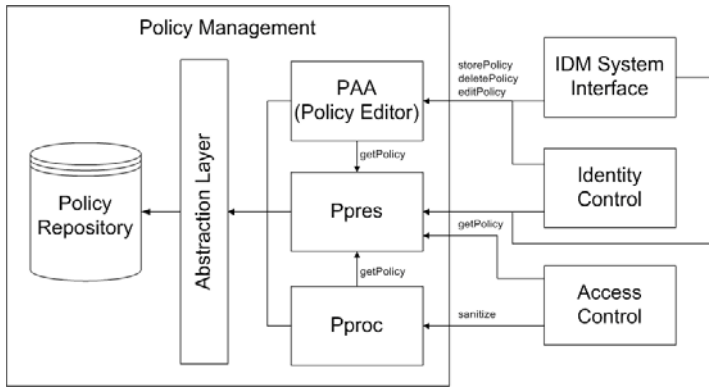
Fig. 14.3 High-level overview of *PM* interactions with other components

the *AC* module returns to the counterparts. This avoids the release of sensitive information related to the policy itself. For instance, suppose that the response returned by the access control is 'undefined' because current information is not sufficient to evaluate condition **equal(user**.Citizenship,'Italy'). *PM* could decide to return to the user the response as it is, including **equal(user**.Citizenship,'Italy'), or otherwise it could modify (sanitize) the condition by requesting the user to declare its nationality (e.g., 'give me your citizenship'). This way, the fact that access is restricted to EU citizens is not disclosed. The filtered information is called sanitized information. The definition of sensitive information and sanitization operations are issues managed in cooperation with the *AC* module. To conclude, *Policy Repository* (*PR*) provides policy storage and search functionalities. *PR* is based on the relational database concept and is designed to be independent from the real storage infrastructure. *PR* provides a fine-grained query infrastructure, based on policy constraints (i.e., object, multiple actions, subject, and so on). An *Abstraction Layer* hides low-level details and isolates *PR* from the outside. By default, *PR* works in an asynchronous way, allowing concurrent access to the data. Each request to *PR* is filtered by *Abstraction Layer*, which unifies the interfaces to the *PR* component.

14.4 Policy Evaluation

We now discuss how access control and data handling policies are evaluated together. Given an access request (see Definition 6) where the object can be a service or some *PII* associated with a user, the request is first received by *ACEF* and then evaluated by *ACDF* in two steps as follows.

Step 1. The access request is evaluated against the applicable access control (or release) policies (see Chapter 11 for more details) collected from the Policy Management module. Note that, if no policy is selected, the access is denied (i.e., the default access decision is *deny-all*). The current implementation is based on policies specified in a Disjunctive Normal Form (DNF), meaning that rules inside the policies are ORed and conditions inside the rules are ANDed. After collecting all applicable policies, each policy is evaluated by inserting the policy conditions in a Reverse Polish Notation (RPN) stack. At the end of the process, the system combines the results of each policy evaluation to reach a 'yes', 'no', or 'undefined' access decision. In case of a negative ('no') access decision, the access request is denied, and the process terminates. In case of a positive ('yes') access decision, *ACDF* has to verify whether there exists some restrictions on the secondary use of the requested object (see Step 2). In case of an 'undefined' access decision, the information submitted by the requester is insufficient to determine whether the request can be granted or denied. Additional information is required by communicating filtered queries to the requester. Such requests are called *claim requests*. It is important to highlight that a claim request could contain sensitive information. In this case, a sanitization process is performed before the claim request is sent to the counterpart, to avoid the release of sensitive information related to the policy itself.

Step 2. *ACDF* queries the PM module to retrieve all data handling policies attached to the object of the request. If no data handling policy is applicable to the request, Step 2 is skipped and the access is granted. Otherwise, applicable data handling rules are retrieved by using *action* and *purposes* specified in the access request as keys. For each applicable data handling rule, the system evaluates the conditions specified in the *recipients*, *gen_conditions*, and *prov* fields (see Chapter 11 for more details). A positive decision is taken if all applicable rules are satisfied by the requester.

Finally, *ACEF* enforces the final access control decision, which is generated by ACDF by composing the results of the above steps of evaluation. In case a positive ('yes') access decision is retrieved in both steps, the requested data/service is released to the requester together with corresponding data handling policies. The requester is then responsible for managing the received data/service following the attached data handling policies.

14.5 A Privacy-Aware Access Control System Prototype

We present the Java-based prototypes of the modules that are part of the privacy-aware access control system developed in the context of the European PRIME project. In particular, we discuss technical details of the *ACDF* and *PM* prototypes, which provide a solution to integrate traditional access

control mechanisms with release and data handling policy evaluation and enforcement. The *ACDF* and *PM* prototypes have been designed taking into account the following requirements:

- *ACDF* and *PM* have to present clear, well-defined, and independent interfaces;
- *ACDF* and *PM* have to support multi-threading, to manage multiple requests at the same time;[3]
- *ACDF* has to support composition of different types of policies;
- *PM* has to be independent from the adopted physical storage.

Our prototypes have been further integrated within the PRIME architecture in Figure 14.2, to provide a complete privacy-aware identity management solution.

14.5.1   ACDF Prototype

The *ACDF* component takes an access decision for all access requests directed to data/services, by evaluating all the policies applicable to the requests. *ACDF* has been designed to be thread-safe and then its implementation supports the execution of multiple *ACDF* instances running at the same time, without any interaction among them. After receiving the access request, *ACDF*:

1. retrieves the access control (release, resp.) policies by querying *PM*;
2. evaluates the access control (release, resp.) policies by using a Reverse Polish Notation (RPN) stack, and takes an access decision;
3. collects the data handling policies attached to the target of the request;
4. evaluates the data handling policies by using a Reverse Polish Notation (RPN) stack, and takes a decision;
5. composes the different evaluations to generate a single access decision.

The mechanism used to evaluate the different types of policies is the same and relies on a Reverse Polish Notation (RPN) stack. In particular, the peculiarity of the Reverse Polish Notation is that the operators follow their operands. For instance, a sum between three and four, which is usually written '3 + 4', in Reverse Polish Notation becomes '3 4 +'. Often interpreters of Reverse Polish notation are stack-based, that is, operands and operators are pushed onto a stack, and when an operation is executed, its operands are extracted from the stack and the result of the operator evaluation is then pushed on the stack. The current implementation of our *ACDF* RPN stack supports the following operands for access control/release policies (and implements similar ones for data handling policies).

---

[3] To provide multi-threading support, all the modules that interact with *ACDF* and *PM* should support it.

	Yes	No	Undefined
Yes	Y	N	U
No	N	N	N
Undefined	U	N	U

	Yes	No	Undefined
Yes	Y	Y	Y
No	Y	N	U
Undefined	Y	U	U

(a)   (b)

Fig. 14.4 3-state *and* operator (a) and 3-state *or* operator (b)

- **SEAnd:** it implements the logical *and* between two *3-state* values (Yes, No, Undefined) as shown in Figure 14.4(a);
- **SEOr:** it implements the logical *or* between two *3-state* values (Yes, No, Undefined) as shown in Figure 14.4(b);
- **SEAction:** it implements the evaluation of the *actions* field of the access/release policies;
- **SEEvidence:** it implements the evaluation of a complex statement (e.g., `equal(user.Name.Given,'Alice')`) using certified information of the requester (i.e., *credentials*);
- **SEOperator:** it implements the evaluation of a complex statement (e.g., `greater_than(user.age,18)`) using declared information of the requester (i.e., *declarations*);
- **SEPurpose:** it implements the evaluation of the *purpose* field of the access/release policies;
- **SESubject:** it implements the evaluation of the *subject* field of the access/release policies.

The RPN stack-based evaluation has the main advantages of being very fast and of making the evaluation process independent from policy syntax and semantics. The translation from each policy language (both access/release and data handling languages) to the RPN format is made by the specific implementation of the *PolicyLoader* interface depicted in Figure 14.5. In particular, the *DHPolicyLoader* class interprets the syntax of DHPs, while the *ACPolicyLoader* class interprets the syntax of access/release policies. In this way, it is possible to add new policy languages by implementing the specific loading class, with a minimal impact on the current implementation.

To conclude this overview of the *ACDF* prototype, it is important to remark that *ACDF* supports conditions to be evaluated both on certified data, issued and signed by authorities trusted for making the statement, and uncertified data, signed by the data owner itself. The declared and certified information relevant to the evaluation process is retrieved from the *Request Context* module. In case of certified information, the *Request Context* module retrieves the information needed by *ACDF* by using the evidence specified within the statement of the policy to be evaluated. For instance, the following XML fragment requires a user with age greater than eighteen. The age attribute has to be certified (evidence) with an *identity document* released by the *Italian Public Administration*.
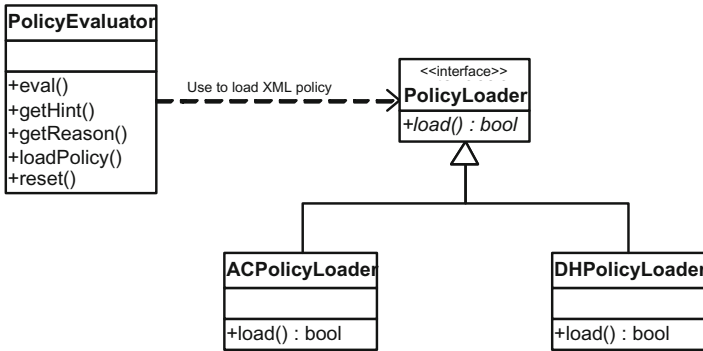
Fig. 14.5 Policy loader infrastructure

```
<group>
 <condition name="greater">
  <argument isLiteral="false">age</argument>
  <argument isLiteral="true">18</argument>
 </condition>
 <evidence>
  <issuer>ItalianPublicAdministration</issuer>
  <proofMethod>X.509</proofMethod>
  <type>identity-document</type>
 </evidence>
</group>
```

14.5.2 PM Prototype

The *PM* module provides functionalities for policy administration. However, considering a privacy-aware access control, the *PM* module accomplishes two main tasks: *i)* it provides a searching engine to retrieve applicable policies to a given request, and *ii)* it provides sanitization functionality.

As depicted in Figure 14.3, the policy search engine is based on an *Abstraction Layer* that provides generic interfaces for accessing *Policy Repository* (*PR*). The *Abstraction Layer* is designed to support multi-threading access to the physical policy storage. The multi-thread supported in *PR* is based on the *Simple Concurrent Object Oriented Programming* (*SCOOP*) model [Mey93], where the concept of thread is extended to the concept of *active object*.[4] To support the *SCOOP* model, *PR* is designed as a singleton *Consumer* [GHJV95]. The Singleton pattern ensures that a class has only one instance and provides a global point of access to that instance. Note that the Singleton

---

[4] Differently from traditional thread concept, *all* the methods of an *active object* run in the separate thread.

pattern can be extended to support access to an application-specific number of instances. *PR* then runs in a separate thread and waits asynchronously for the requests that a set of *Producers* insert in the *PR*'s request queue. Each *Producer* registers a callback method within the request. Finally, *PR* processes the requests one-by-one, and the results are returned to the original requesters.

The *PR* has been designed to support different kinds of repositories. In the current implementation two storage engines are supported:

- **MySQL** [MyS07]: it represents the world's most popular open source database because of its consistent fast performance, high reliability, and ease of use.
- **HSQLDB** [hsq07]: it is a leading SQL relational database engine written in Java. It has a JDBC driver and supports a rich subset of *ANSI-92 SQL* plus *SQL 99* and *2003* enhancements. It offers a small, fast database engine which gives both in-memory and disk-based tables, and supports embedded and server modes.

New engines can be added, if necessary, by implementing the *StorageDriver* interface. This interface works as a *Facade* class on the physical storage, assuring the following basic functionalities: policy addition, policy update, policy deletion, and policy searching.

Focusing on sanitization, *PM* gives a thread-safe process, which provides filtering functionalities on the response to be returned to the counterpart to avoid release of sensitive information related to the policy itself. In particular, it obfuscates conditions in the response to be returned to the counterpart as the access response, according to three possible levels of sanitization: *i) strong sanitization* meaning that a full sanitized condition is generated (e.g., given the original condition '`equal(`**user**`.Name.Given,`'Alice')', a request for declaring **user**.Name.Given is returned); *ii) weak sanitization* meaning that a partial sanitized condition is returned (e.g., given the original condition '`greater_than(`**user**`.Age,18)`', a request for declaring **user**.Age together with the applied operator is returned); *iii) no sanitization* meaning that the full, un-sanitized condition is returned (e.g., '`equal(`**user**`.Citizenship,`'Italy')' is sent to the counterpart as it is).

14.6 Performance Analysis

In this section, we analyze the performance of our access control prototype based on experimental data coming from the testing of the PRIME integrated prototype. The testing has been performed using two different system configurations. The first system configuration (**A**) used a desktop PC with a 3GHz *Intel Core 2 Duo E6850* processor, 4GB of RAM, a *SATA2* disk interface combined with two disks at 7200rpm, and *MySQL* version 5.0.45 installed. The second system configuration (**B**) used a notebook PC with a 2GHz *Intel*
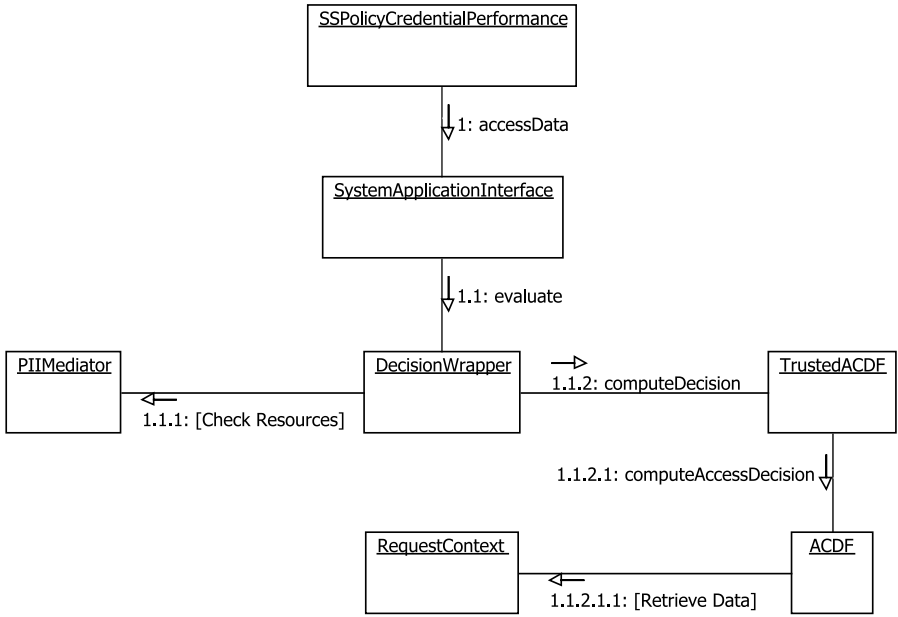
Fig. 14.6 High-level comunication diagram of the evaluation flow

Centrino Duo T2500 processor, 2GB of RAM, a *SATA* disk interface combined with one disk at 7200rpm, and *MySQL* version 5.0.37 installed.

For each system configuration, two batteries of tests have been defined and executed. To minimize spurious cases three runnings for each battery have been performed and their results have been aggregated using the average function. The first battery (battery 1) used an initially empty database to store *PII* and *policies*, while the second one (battery 2) used the database created from the first battery. Each test has then performed 1000 cycles of access requests to resources and, at each cycle, one new policy and one new credential have been created and stored in the MySQL database. Finally, the tests have assumed the scenario where each request is evaluated to 'yes' and the access is granted.

14.6.1   The Evaluation Flow

We provide a performance analysis based on the evaluation flow depicted in Figure 14.6. In particular, the test class requests an access to a resource by calling the *SystemApplicationInterface.accessData()* method. This method forwards the call to the *DecisionWrapper.evaluate()*, and then manages the *DecisionWrapper* result to support interaction with the counterparts. The *DecisionWrapper* component checks if the resource is stored in the *PII* database via *PIIMediator*, which is the component responsible for the management of

the *PII* stored in the database. If the resource is in the database, *Decision-Wrapper* routes the access request to the *ACDF* component[5] preparing the *Request Context* module needed by *ACDF* itself. The *ACDF* component is then responsible for evaluating the policies. To accomplish this task, *ACDF* interacts with the *Request Context* module to retrieve all information needed during the evaluation.

To provide a reliable estimation of the time spent during the evaluation, a number of probing points have been added to measure the following parameters:

- time spent by the *DecisionWrapper* component;
- total time spent by the *ACDF* component to evaluate the policies;
- time spent by the *ACDF* component to evaluate *access/release policies* only;
- time spent during the *access/release policy* evaluation to retrieve the information needed for evaluation, that is, the time spent to access to *Request Context* module;

The above measurements are used in conjunction with the performance data retrieved by *SSPolicyCredentialPerformance* class[6] to measure the total time required to generate a *positive* evaluation response.

14.6.2   Performance Results

Figures 14.7 and 14.8 show the results of our experiments when a *positive* evaluation is reached, and battery 1 and battery 2 are used, respectively. In particular, they show the average percentage of evaluation time spent: *i)* before the access request is sent to *DecisionWrapper*; *ii)* in the *DecisionWrapper.evaluate* method; *iii)* in *DecisionWrapper*; *iv)* in the *TrustedACDF.computeDecision* method (with respect to the total time); *v)* in the *TrustedACDF.compute Decision* method (with respect to the quantity measured at point *ii)*).

By a first analysis of the results shown in Figures 14.7 and 14.8, it comes with no surprise that independently from the batteries of tests under evaluation:

- different amount of RAM used in the two systems configurations do not influence the evaluation performance;[7]

---

[5] Note that, the *DecisionWrapper* does not call directly the *ACDF* component, but the call is wrapped in the *TrustedACDF* class.

[6] *SSPolicyCredentialPerformance* is the test class developed by *BluES'n* application prototype (see Chapter 24 for more details) and used as a starting point in our experiments.

[7] This is probably due to the fact that the testing environment is Windows-based, and then the *Java Runtime Environment* does not take advantages from amount of RAM greater than 2GB.
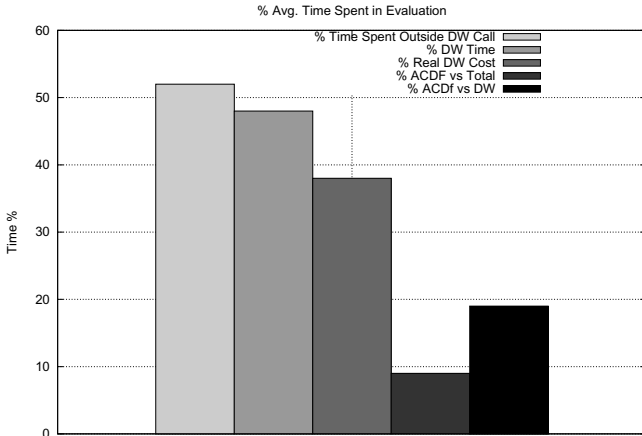
(a)
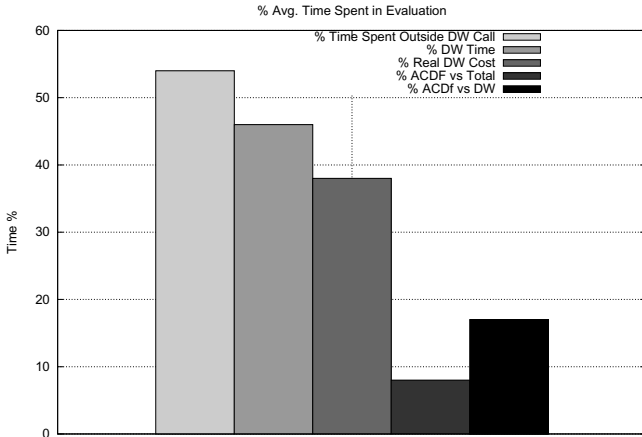
(b)

Fig. 14.7 Average percentage of evaluation time used during the test assuming an initially empty database (battery 1) for the reference system A (a) and B (b)

- different *MySQL* configurations (using the caching support or increasing the quantity of RAM used as cache) seem to do not impact much on the overall test performance. Some empirical tests show an improvement around the 3-5% with respect to the total evaluation time;
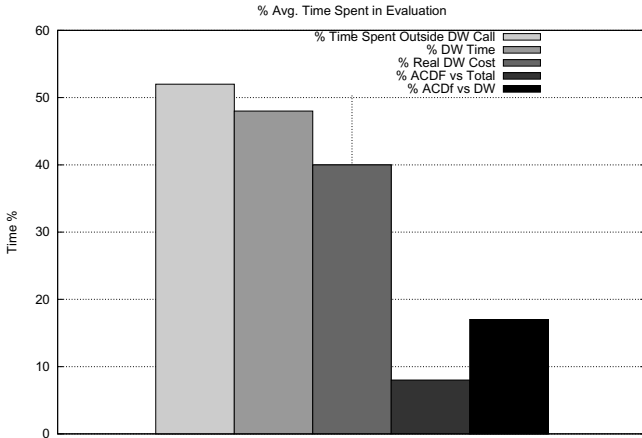
(a)
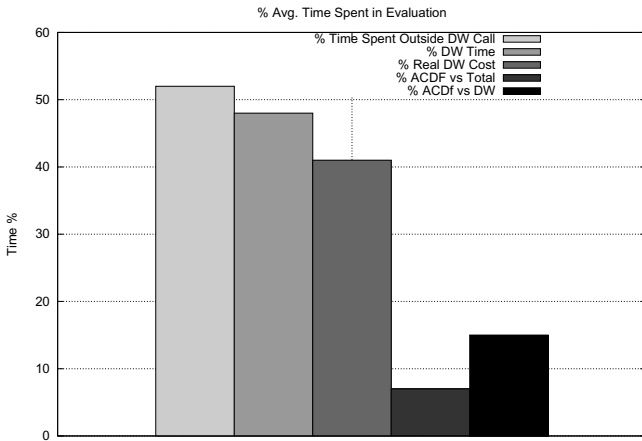
(b)

Fig. 14.8 Average percentage of evaluation time spent during the test assuming an existing database (battery 2) for the reference system A (a) and B (b)

- *SATA2* disk interface seems to guarantee a measurable improvement of the performance; *SATA2* interface in fact should guarantee a doubled bandwidth with respect to *SATA* one.

In more detail, the experimental results in Figure 14.7, which used an initially empty database (i.e., battery 1), show that the distribution of the times over the different components is independent from the system configuration used to perform the analysis. By contrast, the real time spent in the two system configurations is very different. Specifically, the average of the total time spent by system A (Figure 14.7(a)) is 9.1s, with minimum and maximum times 1.0s and 17.7s, respectively, while the average of the total time spent by system B (Figure 14.7(b)) is 24.9s, with minimum and maximum times 3.5s and 53.6s, respectively.

Also the experimental results in Figure 14.8, which used the database filled in the battery 1 experiment (i.e., battery 2), show that the distribution of the times over the different components is independent from the system configuration used to perform the analysis, while the average of the total time spent is very different in the two systems. In particular, the average of the total time spent by system A (Figure 14.8(a)) is 23.5s, with minimum and maximum times 18.4s and 42.8s, respectively, while the average of the total time spent by system B (Figure 14.8(b)) is 66.4s, with minimum and maximum times 50.6s and 101.0s, respectively. In both cases, the differences are due to the different hardware characteristics of the two system configurations.

Comparing the outcomes of the two batteries of tests, it results that the time needed for the overall evaluation flow depends mainly on the database status, that is, how much *PII* data and how many *policies* are stored.

To conclude our performance analysis, we analyze the time required by the *ACDF.computeDecision* method for taking an access control decision. As shown in Figures 14.7 and 14.8, the time used by *ACDF* to compute an access decision represents a minimal part of the time taken by the overall evaluation flow. Also, most of the time required by *ACDF* is used to retrieve the information needed to perform the evaluation, from the *Request Context* module. In our tests, in fact, the access to *Request Context* module consumes the 90-95% of the total time required by *ACDF*. It is important to highlight that, as policies become more and more complex and the number of credentials increases, the time spent to evaluate the request is likely to increase with respect to the time necessary for retrieving the needed credentials.

14.7 Conclusions

In Chapter 11, we have defined an access control model and language for restricting access to resources/data managed by a service provider and release of PII managed by the users, and a data handling model and language

allowing the users to pose restrictions on secondary use of their private data. Here, we have described the prototypes of the components, and the overall architecture providing functionalities for integrating access control/release and data handling policy evaluation and enforcement.

15

Privacy-Aware Identity Lifecycle Management

Marco Casassa Mont

HP Labs

15.1 Privacy-Aware Identity Lifecycle Management: Principles and Concepts

Privacy-aware identity lifecycle management processes must be put in place by enterprises to effectively manage the lifecycle of personal and confidential information according to privacy (law) requirements – over time and across various contexts and solutions. As anticipated, this includes dealing with data retention, data deletion, satisfying notice requirements, supporting data transformations and management of complex workflows. Privacy obligation policies can be used to express these expectations and also take into account user preferences.

This requires a well-planned, systemic and ongoing effort, because: privacy obligation policies and personal preferences can change over time; data and confidential documents can be subject to different privacy and data protection laws depending on geographical and organisational boundaries; data needs to be disposed or transformed over time. The lifecycle of the involved privacy policies must be managed as well.

15.1.1 Obligation Management Framework

As anticipated in Chapter 12 on privacy models and languages, an Obligation Management Framework is introduced to explicitly handle privacy obligations.

In our vision, at the core of privacy-aware identity lifecycle management solutions there is an Obligation Management Framework to centralise (within enterprises) the representation and management of privacy obligations and orchestrate their overall enforcement and monitoring by leveraging

and extending current enterprise IT solutions, in particular Identity Management solutions. Figure 15.1 shows the conceptual model underpinning this framework.
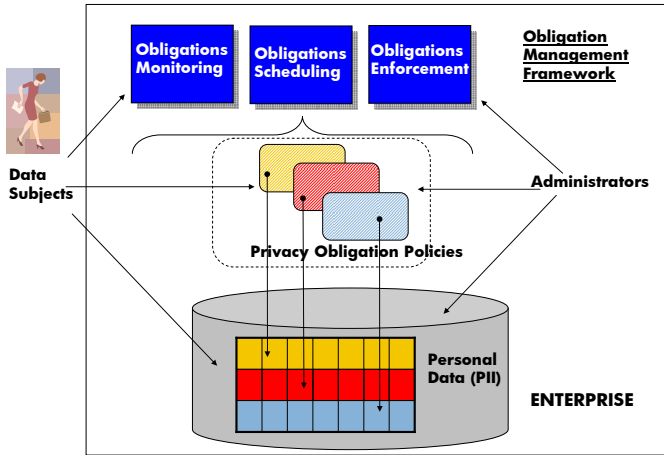
Fig. 15.1 Proposed privacy obligation management model

In our model, privacy obligations are independent entities that are explicitly modeled and managed to enable a privacy-aware lifecycle management of personal data. They are not subordinate to access control aspects. Data subjects can define privacy obligations and associate them to their personal data at the disclosure time (e.g. during a self-registration process) or at any subsequent time. Enterprise privacy administrators can also associate additional privacy obligations, for example dictated by laws or internal guidelines. In our model, the obligation management framework handles these obligations and their associations to personal data by providing the following core functionalities:

- Explicit modeling and representation of privacy obligations: a language/format is defined to explicitly represent privacy obligations in order to analyse them and reason about their implications;
- Scheduling the enforcement of privacy obligations: the system schedules which obligations need to be fulfilled and under which circumstances (events);
- Enforcing privacy obligations: the system enforces privacy obligations once they are triggered. The enforcement ranges from the execution of simple actions to complex workflow involving human interventions;

- Monitoring the fulfilment of privacy obligations: the system monitors and audits the enforced obligations, at least for a predefined period of time, to ensure that the desired status of data is not violated and to report anomalies;
- Administration and lifecycle management of privacy obligations.

These functionalities can be accessed by enterprise privacy administrators and potentially by data subjects, for example to monitor their personal data and check for privacy compliance.

15.2   Obligation Management System

At the very core of this obligation management framework there is an Obligation Management System, in charge of dealing with the enforcement and monitoring of privacy obligations and interacting with other components, such as the privacy-aware access control component.

15.2.1   Design Rationale

The design rationale behind our obligation management system is dictated by the requirements and issues described in Chapter 12 on privacy models and languages and based on our privacy obligation model and obligation management framework.

As previously anticipated, in our approach privacy obligations are handled in an explicit way, independent of and not subordinate to access control. This is required in order to deal with privacy obligations that involve deletion of data, notifications or complex workflows, requests for authorizations and executions of workflows that must be triggered independently by access control activities. Based on this, our design choices reflect the following core aspects:

1. Privacy obligations are self-standing policies, represented with an appropriate language, separated from access control policies;
2. The obligation management system must explicitly parse, manage, schedule, enforce and monitor privacy obligations via dedicated modules. In particular, the monitoring of enforced obligations is important to ensure that the overall system is compliant to enforced privacy obligations and that violations are spotted and reported to administrators.

The fact that the obligation management system must handle privacy obligations over long periods of time and must be always available has also influenced our design choices: survivability and reliability are core requirements. The current design of the obligation management system takes these requirements into account: it is possible to create multiple distributed instances of the obligation management system and monitor for their availability.

15.2.2 System Architecture

Figure 15.2 shows a high-level architecture of an obligation management system supporting the explicit management and enforcement of privacy obligations.
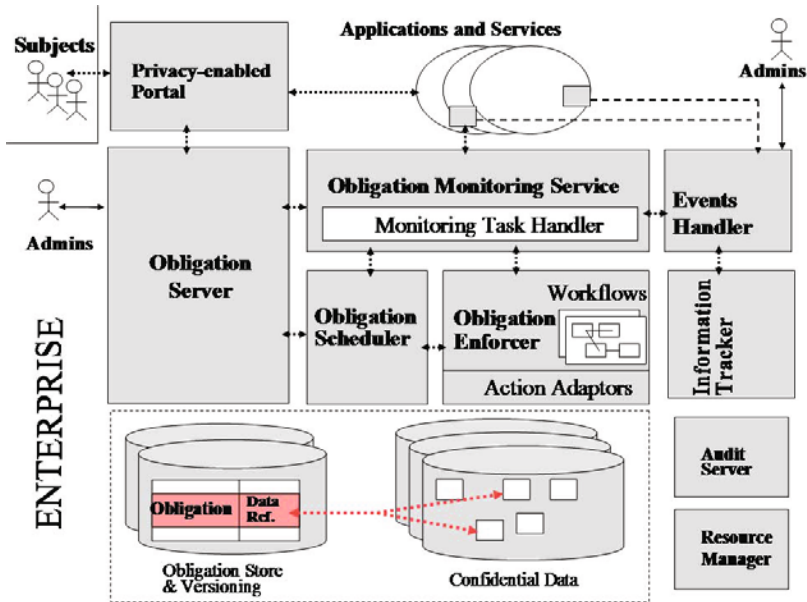
Fig. 15.2 High-level architecture

This obligation management system consists of the following modules:

- **Obligation Server:** it deals with the authoring, management and storage of obligations. It explicitly manages the association of privacy obligations to confidential data and their tracking and versioning. It pushes active obligations (i.e. obligations to be fulfilled) to the Obligation Scheduler. One or more obligation servers can be deployed (and synchronised), depending on needs;
- **Obligation Store and Versioning:** it stores obligations and their mapping to confidential data. Multiple versions of obligations can also be stored in this system, though in the current version of the system this functionality has not yet been implemented;
- **Obligation Scheduler:** it is the module that knows which obligations are active, ongoing obligation deadlines, relevant events and their association to obligations. When events/conditions trigger the fulfilment of one or more obligations, this component activates the correspondent "workflow

processes" of the Obligation Enforcer that will deal with the enforcement of the obligation;

- **Obligation Enforcer:** it is a workflow system containing workflow processes describing how to enforce one or more obligations. The enforcement can be automatic and/or could require human intervention, depending on the nature of the obligation. It is extensible via plug-ins, each of them providing a specific enforcement functionality;
- **Events Handler:** it is the module in charge of monitoring and detecting relevant events for privacy obligations and sending them to the obligation scheduler. The detection of events can happen via instrumented applications/services. They can also be directly generated by users, administrators, the Obligation Monitoring Service and the information tracker;
- **Obligation Monitoring Service:** it is the module, orthogonal to the scheduling and enforcement systems, that monitors enforced obligations by analysing and checking for the effects of their actions, i.e. if the personal data targeted by the obligation is in the desired state;
- **Information tracker:** it is a module that focuses on intercepting events generated by data repositories, databases and file systems containing confidential data and providing this information to the event handler. It is aware of the location of confidential data (as described by the obligation policies) and checks for movements and changes happening to this data;
- **Audit Server:** it audits the relevant events and information generated by the overall system modules and involved applications/services;
- **Resource Manager:** it is a module in charge of checking that all the other system components are running and allocating their services to requestors.

The core "run-time" functionalities provided by a system based on this architecture include:

- **Setting a new privacy obligation** (Figure 15.3): a new obligation is sent to the Obligation Server, either by a data subject or an administrator. The Obligation Server parses and checks for its format correctness. It stores this obligation in the obligation database and communicates it to the Obligation Scheduler to ensure that the obligation will be processed at the due time;
- **Enforcing a privacy obligation** (Figure 15.4): the Obligation Scheduler listens to managed events sent by the Event Handler and checks if any of them (or any combination of them) triggers one of the managed obligations. Should this happen, the Obligation Scheduler communicates with the Obligation Server to retrieve all the relevant information and sends the obligation to the Obligation Enforcer. The Obligation Enforcer analyses the "action part" of the obligation and executes all the listed actions. Independent of the enforcement result, it sends a copy of the obligation to the Obligation Monitoring Service;

- **Monitoring an enforced privacy obligation** (Figure 15.5): the Obligation Monitoring Service periodically checks the status of personal data, against related privacy obligations that have been enforced. This is important for compliance reasons, to identify possible violations or technical problems. For example, in case of deleted data (as a consequence of enforcing an obligation) this module will check if data are actually deleted, for a predefined period of time. It might happen that, because of wrong database synchronisation or back-ups, deleted data reappears in the repository: our system will be able to spot this anomaly.
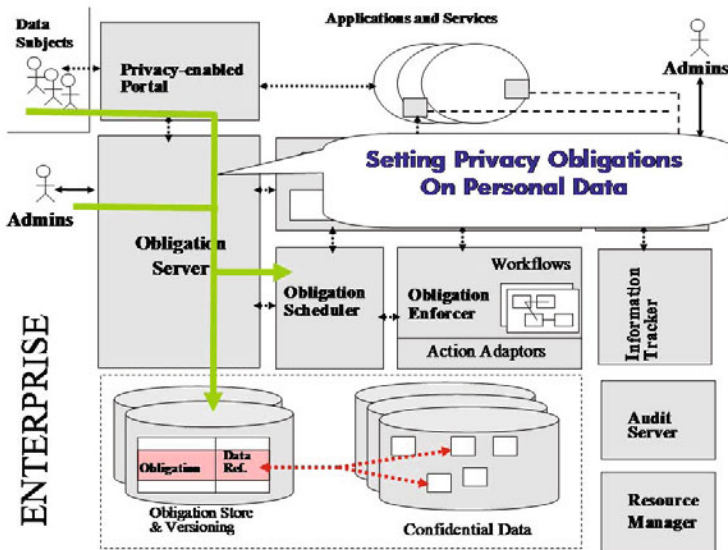
Fig. 15.3 Setting a new privacy obligation

The privacy obligation management system is a critical system: it must survive faults and excessive workloads. Our system has been built to be distributed and the instantiations of its components can be replicated. Multiple distributed instances of all the above components can be created and run in parallel: all of them are stateless, as the relevant information on managed privacy obligations is stored in a replicated database. A (replicated) Resource Manager module manages these instances and allocates these resources to requesters (for example the Obligation Server trying to connect to an Obligation Scheduler or the Obligation Scheduler trying to connect to an Obligation Enforcer).
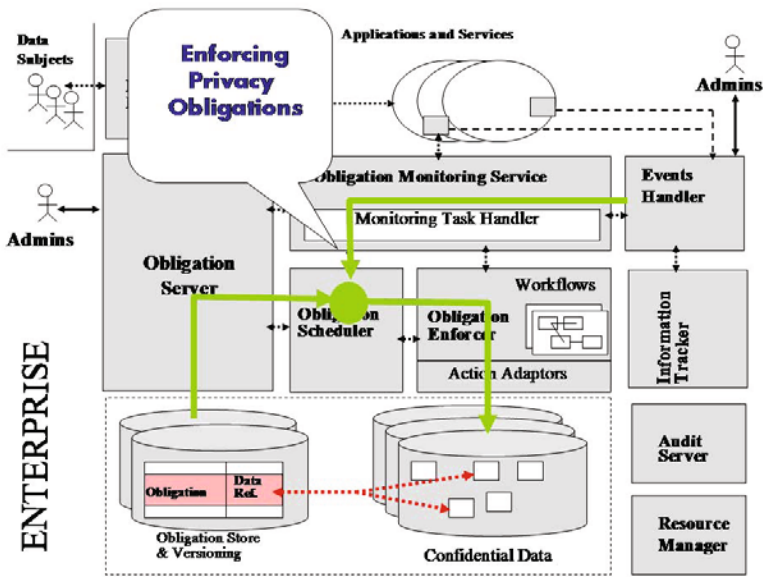
Fig. 15.4 Enforcing a privacy obligation
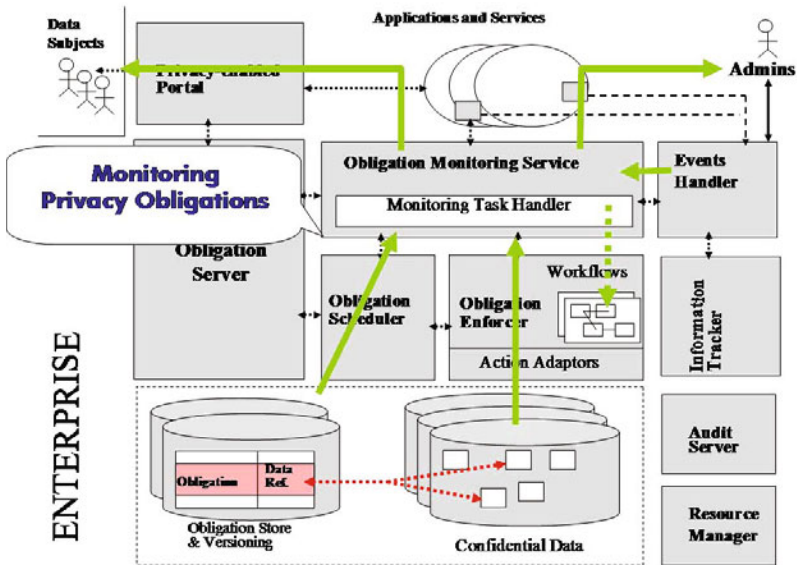
Fig. 15.5 Monitoring a privacy obligation

15.2.3   Implementation Details

This section provides more technical details of the architecture of the obligation management system along with a description of its internal modules, data structures and related interactions, as implemented in a prototype at HP Labs, Bristol, UK. Our prototype has been integrated with external components provided by PRIME partners, in an integrated prototype. Some of our technical choices have been dictated and constrained by this. We will refer to this integrated prototype as the "integrated PRIME prototype".

Figure 15.6 provides a view of the internal modules implemented in the current version of the obligation management system prototype.
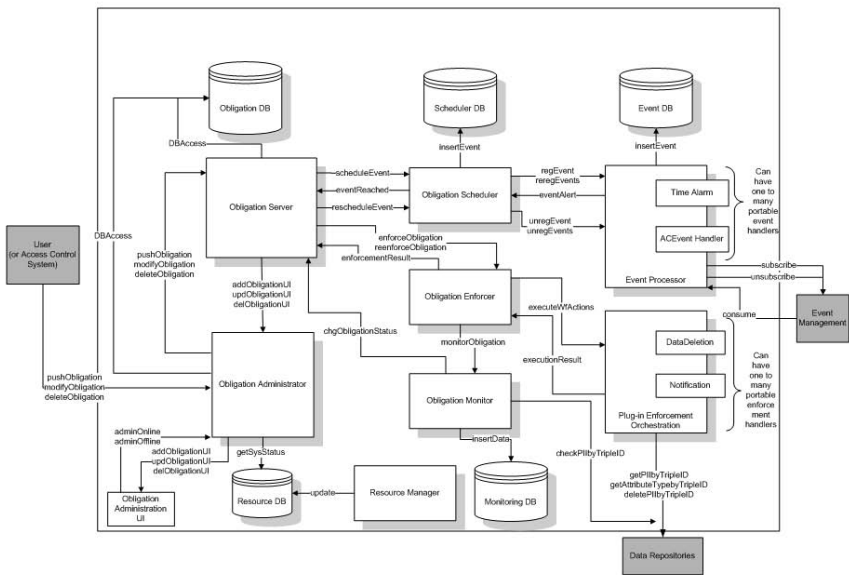
Fig. 15.6 Details of PRIME obligation management system

All these modules have been implemented in Java, as RMI objects. A description of the main modules along with their APIs/interfaces follows.

15.2.3.1   Obligation Administrator

The Obligation Administrator is the module that implements the external-facing functions provided by the obligation management system in order to

make them accessible by other external components. It is in charge of interacting with the Obligation Server to coordinate the overall management of privacy obligations within the obligation management system. It also implements internal functions to interact with the Obligation Administration UI and support its management tasks, currently limited to the visualization of active and monitored obligations. The Administration UI mainly interacts with this module. Its key functions are:

- **pushObligation:** This function is invoked by an external component to push a new privacy obligation to the system. It passes this new obligation to the Obligation Server for its processing.
- **modifyObligation:** This function is invoked by an external component to modify an existing privacy obligation in the system. It passes this modified obligation to the Obligation Server for its processing.
- **deleteObligation:** This function is invoked by an external component to delete an existing privacy obligation in the system. It passes this information to the Obligation Server for its processing.
- **addObligationUI:** This function is invoked by the Obligation Server to notify the Obligation Administrator that a new obligation (that has been fully processed by the Obligation Server) must be shown by the Administration UI.
- **updObligationUI:** This function is invoked by the Obligation Server to notify the Obligation Administrator that an updated obligation (that has been fully processed by the Obligation Server) must be shown by the Administration UI.
- **delObligationUI:** This function is invoked by the Obligation Server to notify the Obligation Administrator that a deleted obligation (that has been fully processed by the Obligation Server) must be removed from the Administration UI.
- **adminOnline:** This function is invoked by the Administration UI to notify that it is currently online, i.e. it can receive messages about obligations to refresh the displayed information.
- **adminOffline:** This function is invoked by the Administration UI to notify that it is currently offline, i.e. it cannot receive messages about obligations.

It is important to notice that in the context of the integrated PRIME prototype, privacy obligations (associated to personal data) are also known by other PRIME components, external to the obligation management system. In particular, a copy of these obligations is stored along with personal data in the PRIME "data repository": this because of the data model we chose to implement in PRIME, to guarantee an initial degree of "stickiness' of privacy obligations to data. Because of this, no "getObligation" method is required to retrieve obligations from the obligation management system as external components can obtain this information directly from the PRIME data repository. This aspect is also reflected by other internal modules of our system. At the

time of writing this chapter, the integrated PRIME prototype also does not support modifications and deletions of privacy obligations. This capability will be provided in a future version of this prototype.

15.2.3.2 Obligation Server

The Obligation Server is the module in charge of processing and handling obligations. It controls all the aspects involving the lifecycle management of privacy obligations: it deals with the local storage of privacy obligations and the coordination of the scheduling of obligations' events and the enforcement of obligations' actions. It also keeps an up-to-date registry of the enforced obligations, based on notifications coming from the Obligation Monitor component. Its key functions are:

- **pushObligation:** This function is invoked by the Obligation Administrator to push a new privacy obligation to the Obligation Server. The Obligation Server creates a unique identifier for the new obligation, stores it into the local Obligation Database, notifies the Administrator UI (by invoking the Obligation Administrator's addObligationUI function) about this obligations and interacts with the Obligation Scheduler to set the events relevant to trigger this obligation.
- **modifyObligation:** This function is invoked by the Obligation Administrator to modify a privacy obligation.
- **deleteObligation:** This function is invoked by the Obligation Administrator to delete a privacy obligation.
- **eventReached:** This function is invoked by the Obligation Scheduler to notify the Obligation Server that an obligation has to be enforced, as the events relevant to trigger the obligation have happened. The Obligation Server interacts with the Obligation Enforcer to ensure that the relevant obligation's actions are executed.
- **enforcementResult:** This function is invoked by the Obligation Enforcer to notify the Obligation Server about the result of enforcing an obligation. The Obligation Server updates the obligation status within the local database and notifies the Administrator UI.
- **chgObligationStatus:** This function is invoked by the Obligation Monitor to notify that the status of an obligation has changed (i.e. it has been violated or it is OK).

15.2.3.3 Obligation Scheduler

The Obligation Scheduler is the module in charge of scheduling events associated to obligations. These events could be time based (i.e. a specific date and time), access based (i.e. related to access events generated when accessing specific personal data) or based on counters (i.e. the value of an access counter has reached a predefined value). Events could be "ongoing", i.e. they

occur periodically (e.g. every month). Events could be composed in logical expressions, involving AND and OR compositions of other events. The current system handles time-based events, counter-based events and ongoing events and their AND/OR compositions. The NOT operator is not yet explicitly supported, as it is not required to handle the current managed set of privacy obligations: it will be introduced in a future version of our system. Its implications on events and complex events need to be fully explored: it is going to be part of our future research activities. This module processes incoming events, forwarded by the Event Processor, and checks if any of them triggers a managed obligation. In case it does, it notifies the Obligation Server, to ensure the enforcement of the relevant obligation. Its key functions are:

- **scheduleEvent:** This function is invoked by the Obligation Server to schedule an event associated to an obligation. This event could be composite, i.e. a logical AND/OR logical expression of other events. The Obligation Scheduler parses this event, decomposes it into simple events (if the event is a composite one) and stores all this information in the local obligation database. For each simple event it sends related information to the Event Processor, in order to be notified once the event happens. When an event happens, this event is no more considered as active and needs to be rescheduled if its activity needs to be rescheduled.
- **rescheduleEvent:** This function is invoked by the Obligation Server to reschedule an event associated to an obligation. This happens when the obligation is an ongoing obligation, hence events need to be scheduled on an ongoing basis (e.g. once a month). It executes the same activities done by the scheduleEvent function.
- **eventAlert:** This function is invoked by the Event Processor to notify the Obligation Scheduler that a relevant event (previously set by the Obligation Scheduler) has happened. The Obligation Scheduler processes this event and checks if it triggers any managed obligation. In this case, it will interact with the Obligation Server to ensure that the obligation is enforced.

15.2.3.4   Event Processor

The Event Processor module is in charge of processing simple events that are relevant to the obligation management system, to triggering managed events. Based on requests for handling events sent by the Event Scheduler, the Event Processor interacts with any external Event Management component to subscribe (or unsubscribe) for related event notifications (the Event Management component is an abstraction of external components that generate events. Its detailed functionalities are not described as it is beyond the scope of this work). This in particular happens for access control-based events or events related to other components. The Event Processor uses a sub-module, called TimeAlarm, to generate time-based events. Its architecture is extensible via

plug-in sub-modules, each of them being in charge of receiving and processing specific types of events. In addition to the TimeAlarm plug-in, the current implementation provides an ACEventHandler plug-in, to handle access control related events. The key functions of the Event Processor are:

- **regEvent:** This function is invoked by the Obligation Scheduler to notify the Event Processor about the need to handle a specific type of event. In case of time-based event, the Event Processor will internally register the interest for this event, and at the right time will notify the Obligation Scheduler of its occurrence. In case of access control and other events, the Event Processor will subscribe for this type of events (if it has not yet done it in the past) by interacting with the Event Management component. It will also locally register its interest for this event.
- **reregEvents:** This function is invoked by the Obligation Scheduler to register again for one or more events, in case of ongoing obligations. The relevant events are already known by the Event Processor but some of their parameters might have changed (for example the triggering time, in a time-based event). This function allows the Event Processor to update parameters associated to existing event records, to enable the management of ongoing obligations.
- **consume:** This function is invoked by the Event Management component to notify the Event Processor that an event (for which it registered its interest) has occurred. The event passed as a parameter is processed by the Event Processor and sent to the Obligation Scheduler for further processing (such as its evaluation in the context of logical expressions, involving multiple events). It could trigger the enforcement of one or more privacy obligations.
- **unregEvent:** This function is invoked by the Obligation Scheduler to un-register its interest for a particular type of event.
- **unregEvents:** This function is invoked by the Obligation Scheduler to un-register its interest for a set of types of events. This happens in case of ongoing obligations, that have been fully enforced (i.e. do not need to be further processed by the system).

15.2.3.5 Obligation Enforcer

This module is in charge of enforcing privacy obligations, i.e. executing actions as defined within privacy obligations once these obligations have been triggered by relevant events. The Obligation Enforcer module is notified by the Obligation Server about the need to enforce obligation actions. These actions might involve the deletion of personal data or part of personal data, sending notifications or handling counters. For ongoing obligations, related actions might need to be periodically enforced (for example resending notifications every month). The Obligation Enforcer interacts with the Plug-in Enforcement Orchestration module to enforce these actions and communicates the outcome to the Obligation Server. It also notifies the Obligation

Monitor component about the need of monitoring enforced obligations. Its key functions are:

- **enforceObligation:** This function is invoked by the Obligation Server to notify the Obligation Enforcer module about the need to enforce an obligation, i.e. to execute the actions specified by this obligation. The Obligation Enforcer stores a record in the local database and interacts with the Plug-in Enforcement Orchestration module to execute these actions. It returns the result of the enforcement activity to the Obligation Server.
- **reenforceObligation:** This function is invoked by the Obligation Server to notify the Obligation Enforcer module about the need to re-enforce an ongoing obligation, i.e. to execute again the actions specified by the obligation such as notifications. This might require the system to increase local counters, in case ongoing obligations need to be repeated for a predefined number of times. It returns the result of the enforcement activity to the Obligation Server.
- **executionResult:** This function is invoked by the Plug-in Enforcement Orchestration module to notify the Obligation Enforcer about the current status of an enforced obligation. Its status could be OK or there could be a FAILURE (obligation enforcement is unsuccessful). In both cases the Obligation Enforcer notifies the Obligation Monitor that it has to monitor the status of this obligation. For example, if the enforced obligation deleted personal data in the database, the Obligation Monitor checks that these data do not reappear in the database.

15.2.3.6 Plug-In Enforcement Orchestrator

This module is in charge of enforcing specific actions as described by a privacy obligation. This might include the execution of complex workflows, involving the coordination of human interactions. Its architecture is extensible via a plug-in based approach. In the current implementation two core actions can be enforced: deletion of data and notification of users via e-mail. In particular for deletion of personal data, it interacts with external data repositories, specifically an RDBMS database. Its key function is:

- **executeWfActions:** This function is invoked by the Obligation Enforcer to notify the Plug-in Enforcement Orchestrator module about the need to enforce one or more actions. In the current version actions might require the deletion of data and notifications to users. The Plug-in Enforcement Orchestrator analyses the types of actions and orchestrates their enforcement by calling plug-in modules, specialized to enforce specific types of actions. At the moment two plug-ins are implemented: DataDeletion and Notification. The DataDeletion plug-in interacts with the data repository to actually delete data. The Notification plug-in interacts with the data

repository to retrieve the actual e-mail address to send a notification to. The overall enforcement result is returned to the Obligation Enforcer.

15.2.3.7   Obligation Monitor

This module is in charge of monitoring enforced obligations for compliance, i.e. checking that the effect of enforcing obligation actions is not compromised over time (e.g. deleted data that reappears in the database because of wrong database back-ups or synchronizations). The obligations that need to be monitored are specified by the Obligation Enforcer. Periodic notifications about the status of monitored obligations are sent to the Obligation Server. At the moment this monitoring capability is passive, in the sense that it highlights violations but it takes no automatic actions to correct it. Administrators need to explicitly ask the system to re-enforce the violated obligations. In a future version of our prototype the automation of this aspect will be further analysed and implemented. Its current key function is:

- **monitorObligation:** This function is invoked by the Obligation Enforcer to notify the Obligation Monitor module about the need to monitor an obligation. It stores a record in the local database about the obligation to be monitored. In case of obligations involving deletion of data, it periodically interacts with the data repository to verify if the deleted data has not reappeared.

15.2.3.8   Resource Manager

The Resource Manager is the module in charge of managing, at run-time, the actual RMI instances of all the above modules. The obligation management system ensures the provision of a reliable and survivable service, even in case of occasional, localised failures. To achieve this, runtime redundancy is required for all the above critical components to cope with failures and changing workloads. Multiple RMI instances of all the above modules can be created at runtime. This is possible as all these modules can run as self-standing RMI objects and all of them are stateless: they store and share all the relevant information within a local database. In the current configuration, up to three instances of the Resource Manager can run at the same time. Their RMI interface names are well known by all the other modules of the obligation management system: these modules will sequentially try to contact them, until they find a running instance. At the start-up time, each module registers its RMI interface name to the Resource Manager. In case a module wants to interact with another module, it will first interact with the Resource Manager. The Resource Manager returns the interface name of one of the currently available instances of the requested module.

The Resource Manager also periodically checks for the status of all these instances and updates information in a local database: this information is

displayed by the Administrator UI. The functionalities provided by this component are related to "operational" aspects of the obligation management system and affect all the involved modules: the related functions, described below, are not displayed in the architectural diagram. Its key functions are:

- **register:** This function is invoked by any module of the obligation management system to register its RMI interface name with the Resource Manager. Multiple instances of each module might be registered.
- **getResource:** This function is invoked by any module of the obligation management system to get the RMI interface name of another module of the system. If multiple instances are available and running, the Resource Manager will randomly choose one. If no instance is available, this function will fail.

Our current prototype implements a "synchronised" access to and update of the tables stored in the local databases ("Obligation DB", "Scheduler DB", "Event DB", "Resource DB" and "Monitoring DB"), in order to avoid conflicts and inconsistencies: this is achieved by leveraging standard techniques involving the usage of "critical sections" in the Java code.

15.2.4   Interaction Flow

The main interaction flow (involving most of the above modules) is triggered when a new privacy obligation is submitted to the obligation management system. Only the main interaction steps are described. For simplicity, the description of the steps involving refreshing the UI components is omitted:

1. <pushObligation>: the Obligation Administrator gets a privacy obligation from a user or an administrator. It passes it to the Obligation Server;
2. <pushObligation>: the Obligation Server gets a privacy obligation from the Obligation Administrator;
3. The Obligation Server validates the format of the received obligation;
4. If the obligation is invalid, system returns, process ends;
5. The Obligation Server inserts the valid obligation into the "Obligation DB";
6. <scheduleEvent>: the Obligation Server extracts the event block from the obligation, sends the event block to the Obligation Scheduler and (asynchronously) waits for the alert confirming that the event has happened;
7. <insertEvent>: the Obligation Scheduler decomposes the complex event into single events, and inserts them into the "Scheduler DB";
8. <regEvent>: the Obligation Scheduler registers the single events with the Event Processor, and waits for the alert when the event happens;
9. <insertEvent>: the Event Processor inserts the events into the "EventDB";
10. The Event Processor checks each type of the new events;
11. If the event is time based, it will be sent to the Time Alarm (example of time based event: when the time reaches 01/01/2006 12:00);

12. If the event is access control based, it will be sent to the ACEventHandler that will register its interest in this event with an external Event Management component;
13. <consume>: the Event Processor receives the events from Event Management component. Those events provide the access control information (e.g. the credit card number of user uid05 has been accessed);
14. <eventAlert>: the Event Processor gives alerts to the Obligation Scheduler when the registered event happened;
15. <eventReached>: the Obligation Scheduler updates the status of the event in DB according the received alerts. The Obligation Scheduler sends out the "eventReached" acknowledgement to the Obligation Server when all the conditions in a complex event have been fulfilled;
16. <enforceObligation>: the Obligation Server extracts the action block of the obligation from database, and sends the action block to Obligation Enforcer;
17. <executeWfActions>: the Obligation Enforcer decomposes the complex action into single, ready to enforced actions, and then the actions are sent to the Plug-in Enforcement Orchestration;
18. The Plug-in Enforcement Orchestration forwards the action to suitable plug-ins such as Deletion and Notification plug-ins;
19. <executionResult>: the Plug-in Enforcement Orchestration replies with the execution result;
20. <enforcementResult>: the Obligation Enforcer collects the results from the Plug-in Enforcement Orchestration, then sends the enforcement result to the Obligation Server;
21. <monitorObligation>: the Obligation Server extracts the actions from the obligation, and sends it to Obligation Monitor for monitoring;
22. <insertData>: the Obligation Monitor decomposes the complex actions into single actions, and inserts them into Monitoring DB;
23. <chgObligationStatus>: the Obligation Monitor alerts any violation of the monitoring obligations to the Obligation Server.

It is important to notice that this interaction flow involves steps that can happen in an asynchronous way: for example only when a combination of events happens, this triggers the enforcement of related actions. Further research is required to understand the impact of creating and managing large sets of privacy obligations on large databases of personal data: in this context, the management of related events could be critical. The approach based on replicated instances of critical system components could be exploited to address this issue and balance the workload of the Event Processor. Further research and work could also be done to optimise the creation and management of privacy obligations, for example by "automatically clustering" obligations that share the same triggering events, in order to minimise the set of events that must be handled.

15.2.5   Event Management Framework

The obligation management system relies on an external Event Management Framework to receive relevant events and notifications in order to trigger privacy obligations.

As described in the previous sections, the event management model adopted in our system – and pursued in the context of the European PRIME project – is based on a producer/consumer model.

An external event management system is in charge of dealing with registration of producers and consumers and to handle the delivery of generated events. In this context, the obligation management system is just a consumer of events, including:

- **Time-based events**;
- **Access control-based events**;
- **Intrusion detection events**;
- **Context-based events** (system status, changes of configuration, etc.).

For simplicity, in the current version of the prototype time-based events are directly generated by the "TimeAlarm" sub-module within the "Event Processor" module of our prototype. In a future version, time-based events could also be generated by an external time server and consumed by our system.

It is beyond the scope of this document to describe in detail what an Event Management Framework is and how it can be implemented. However we recognise that this framework has important implications and requirements on the underlying IT infrastructure.

At the very base, it requires the instrumentation of data repositories, systems and (potentially) applications and services in order to generate the relevant events.

For example, the instrumentation of data repositories, such as RDBMS databases, to generate events based on accesses of stored personal data, might require the definition, deployment and management of triggers and active rules.

In the integrated PRIME prototype, that (in addition to the obligation management system) includes various components built by other PRIME partners, the access control system is in charge of intercepting attempts to access personal data stored in databases and (along with making access control decision and enforcing them) generating the relevant events.

Whatever approach is used, an infrastructural overhead is generated. This overhead has to be measured and its impact on the infrastructure and systems has to be quantified.

As the obligation management system is orthogonal to the event management framework (as long as it is based on a producer/consumer framework and the semantic of the events is shared with our system), we could leverage event management frameworks already available on the market.

The generation, logging and analysis of events are core functionalities required for IT Compliance Management solutions, in the area of enterprise IT Governance. Products and solutions available on the market already provide their own event management frameworks.

If compatible with our requirements, the event management frameworks provided by these solutions could be leveraged and integrated with our obligation management system to avoid duplication of efforts.

This aspect and the implications of their integration with our obligation management system will be addressed in a next stage of our project.

15.2.6   Data Repository

For operational reasons, the obligation management system stores privacy obligations and related metadata in internal data repositories. In the previous sections of this chapter we logically referred to these repositories as "Obligation DB", "Scheduler DB", "Event DB", "Resource DB" and "Monitoring DB".

In the current implementation, for simplicity, all these repositories are implemented as tables within a unique relational database (in our prototype we used a MySQL database system). The main tables storing this information are:

- Obligations;
- Events;
- Expressions;
- Actions;
- Monitored Items;
- Resources.

A diagram describing the relationships between the above tables is shown in Figure 15.6. A description of the content of each table follows.

15.2.6.1   Obligation Table

This table is the main storage of privacy obligations, formatted as XML strings, and related metadata describing their statuses. The main fields of this table are:

- **ObligationId:** it stores the unique identifier of the privacy obligations;
- **InitTime:** it stores the time when this obligation has initially been sent to the obligation management system;
- **ModifyTime:** it stores the last time when this obligation has been modified;
- **OblType:** it stores the type of obligation (long-term, short-term, ongoing, etc.);
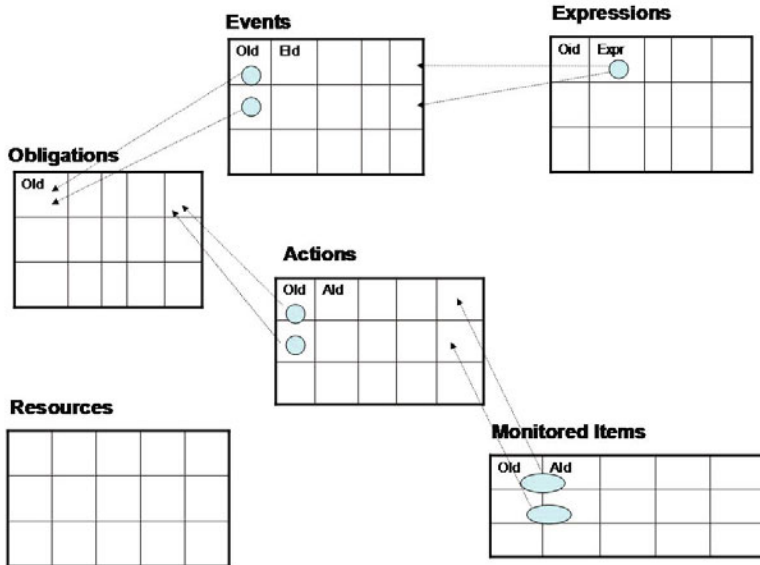
Fig. 15.7 OMS prototype - system tables

- **Description:** it stores a "human readable" description of a privacy obligation, as provided by the administrator that authored the obligation;
- **Obligation:** it stores the entire XML string representing the obligation;
- **Target:** for performance reasons, it stores the "Target" portion of the XML obligation string;
- **Events:** for performance reasons, it stores the "Events" portion of the XML obligation string;
- **Actions:** for performance reasons, it stores the "Actions" portion of the XML obligation string;
- **Status:** it stores the current, up-to-date, status of a privacy obligation (scheduled, enforcing, ok, violated).

15.2.6.2   Event Table

This table contains a list of "simple events" associated to privacy obligations managed by the system. Its content is the result of parsing the "Events" section of each obligation. As a result, multiple event records could be associated to the same obligation. The main fields of this table are:

- **ObligationId:** it is the unique identifier of the obligation an event belongs to;
- **EventId:** it is the unique identifier of an event, in the context of an obligation;

- **EventType:** it classifies the type of managed event (timeout, access, delete);
- **ScheduledNumber:** it contains the number of times this events is expected to happen to trigger the obligation. It is a counter;
- **Status:** it contains the current status of the event (stopped, closed, etc.).

15.2.6.3   Expressions Table

This table refers to events contained in the "Events" table and explicitly describes logical combinations (AND, OR combinations) of these simple events. These logical combinations are derived from the original "Events" sections of privacy obligations:

- **ObligationId:** it is the unique identifier of the obligation an event belongs to;
- **Expression:** it is a string containing a logical combination of simple events. Multiple simple events, defined in the "Events" table, are combined in AND/OR logical expressions, by using their EventId;
- **ScheduledNumber:** it contains the number of times this complex event is expected to happen to trigger the obligation. It is a counter;
- **Status:** it contains the current status of the complex event (stopped, closed, etc.).

15.2.6.4   Action Table

This table contains a list of "simple actions" associated to privacy obligations managed by the system. Its content is the result of parsing the "Actions" section of each obligation. As a result, multiple actions could be associated to the same obligation. The main fields of this table are:

- **ObligationId:** it is the unique identifier of the obligation an action belongs to;
- **ActionId:** it is the unique identifier of an action, in the context of an obligation;
- **Action:** it contains the XML portion describing this action (delete, notify, trigger workflow, etc.);
- **EnfNumber:** it contains the number of times this action has been enforced;
- **Status:** it contains the current status of the action (success, failure, etc.).

The ActionId key is relative to the context of an obligation and unique only within this obligation (i.e. the same key could be used in different obligations). The combination of the ObligationId and ActionId keys ensures the unique identification of an action, within the obligation management system.

15.2.6.5   MonitoredItems Table

This table contains the status of enforced obligations. Specifically the system stores the status of each action of a given enforced obligation. The main fields of this table are:

- **ObligationId:** it stores the unique identifier of the privacy obligations;
- **ActionId:** it is the unique identifier of an action, in the context of an obligation;
- **InitTime:** it stores the time when this obligation has initially been sent to the obligation management system;
- **ModifyTime:** it stores the last time when this obligation has been modified;
- **Action:** it contains the XML portion describing this action (delete, notify, trigger workflow, etc.);
- **Type:** it describes the type of enforced action (delete, notify, etc.);
- **Status:** it stores the up-to-date status of an enforced action (ok, violated).

It is important to notice that also in this table both the ObligationId and the ActionId keys are used to identify an action, for the reasons explained in the "Action Table" subsection.

15.2.6.6   Resources Table

This table contains the information about all the instances of RMI modules of the obligation management system and their statuses. The main fields of this table are:

- **Module:** it contains the type of system module (ResourceManager itself, ObligationServer, ObligationScheduler, ObligationEnforcer, Obligation-Monitor, EventProcessor, EnforcementOrchestrator). Multiple records of the same type could be present, as the system can handle multiple instances of the same components, for fault tolerance and load balancing reasons;
- **Server:** it contains the logical (DNS) name of the server hosting this module;
- **RMIName:** it stores the RMI logical name of the module, used by other module to remotely connect to the object;
- **Status:** it stores the current status of the module (alive, dead).

15.2.7   Administration GUI

The current prototype provides (basic) administrative management functionalities via a graphical Administrative UI. This UI provides the following graphical views:

- **Admin View**;
- **Monitoring View**;
- **System View**.

In the Admin View of this UI, administrators can check and browse for the current set of managed privacy obligations (either to be enforced or enforced) – see Figure 15.8. In this context, it is possible to restrict, in a fine-grained way (based on time intervals), the set of privacy obligations that an administrator wants to investigate.
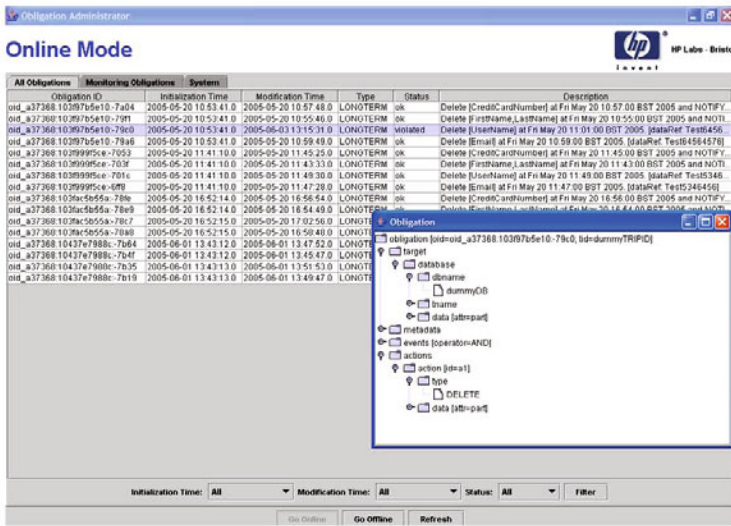
Fig. 15.8 Obligation management: administrative GUI

For each managed obligation, the UI provides the following information:

- **Obligation Id:** it is the unique privacy obligation identifier, used within the entire system;
- **Initialization Time:** it is the time when the privacy obligation has been initially submitted to the system;
- **Modification Time:** it is the last time recorded where the privacy obligation has been subject to any management activity;
- **Type:** it is the type of privacy obligations. The currently-supported types are: "SHORT-TERM", "LONG-TERM", "TRANSACTIONAL", "ON-GOING";
- **Status:** it describes the current, up-to-date, status of the obligation. The currently-supported statuses are: "SCHEDULED', "ENFORCING', "OK", "VIOLATED";

- **Description**: it is a human readable description of the privacy obligations. This information is derived from the metadata associated to the privacy obligations, within the XML format.

It is important to notice that this UI can provide a list of all the managed privacy obligations, whatever their statuses are. However, because this list can be very large, it could be unmanageable. The current UI already provides the administrators with mechanisms to focus on a subset of this list, based on any combination of the following criteria:

- Initialization time of an obligation;
- Modification time of an obligation;
- Status of an obligation.

These filtering mechanisms are made available to the administrators via a few selection fields, available at the bottom of the UI – see Figure 15.8. By double-clicking on any obligation row, the administrator can get a detailed view of the internal components of this obligation, via a pop-up window. This window contains a tree-based representation of the obligation that can be easily navigated – see Figure 15.8.

The **Monitoring View** is based on a similar UI, with exactly the same fields. However this UI provides a graphical view of the status of privacy obligations that have been enforced and that are currently monitored - see Figure 15.9. Each obligation is displayed with an associated colour:

- **GREEN:** the status of the obligation is OK. This means that the data targeted by the obligation is in the expected status, as dictated by the enforced obligations;
- **RED:** the obligation is VIOLATED. This means that the data targeted by the obligation is not in the expected status, dictated by the enforced obligations.

The **System View** provides a system perspective illustrating the current status and availability of the various system modules – see Figure 15.10. For each system module the UI shows the following information:

- **Component:** it is the logical name of a system module (e.g. Resource Manager, Obligation Server, Obligation Enforcer, Obligation Monitor, Event Processor, Enforcement Orchestration). More than one instance of the same name could appear, as each of these modules might be instantiated multiple times, for fault tolerance and load balancing reasons;
- **Server:** it is the name of the server (platform) hosting the instance of the RMI module;
- **RMI Name:** it is the name of the RMI interface associated to the module;
- **Status:** it provided an up-to-date status of the module (e.g. "DEAD" or "ALIVE").
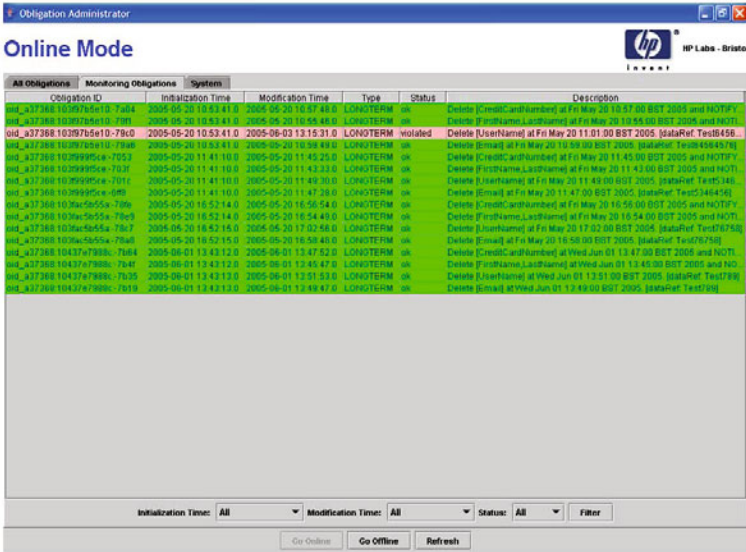
Fig. 15.9 Obligation monitoring: administrative GUI
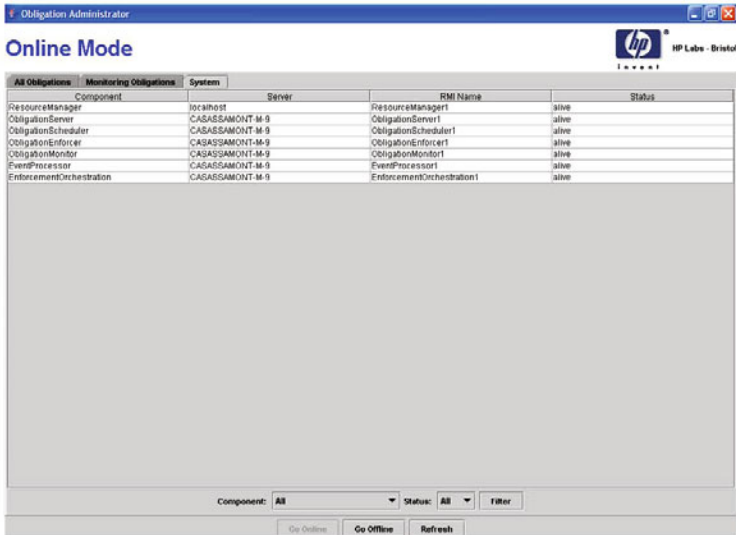
Fig. 15.10 Obligation system monitoring: administrative GUI

15.2.8   Discussion

The obligation management system presented so far in this chapter aims at illustrating the basic, underlying principles and criteria to automatically manage, enforce and monitor privacy obligations. We provided a detailed description about how to build a system in grade of achieving this.

At the same time we are well conscious that this system needs to be scalable, i.e. be able to process a high volume of data and related processing tasks. The following sections provide further details about how this has been addressed in PRIME by the Scalable Obligation Management System.

15.3   Scalable Obligation Management System

The Scalable Obligation Management System (SOMS) [Cas06, CB07b] is an evolution of the Obligation Management System (OMS), presented in the previous section, to interpret, enforce and monitor parametric obligation policies (see Chapter 12 on obligation policies).

15.3.1   Scalable Obligation Management Framework

Parametric obligation policies must be deployed in an obligation management framework for their interpretation, enforcement and monitoring. Figure 15.11 provides a high-level view of the key aspects involved in this process.
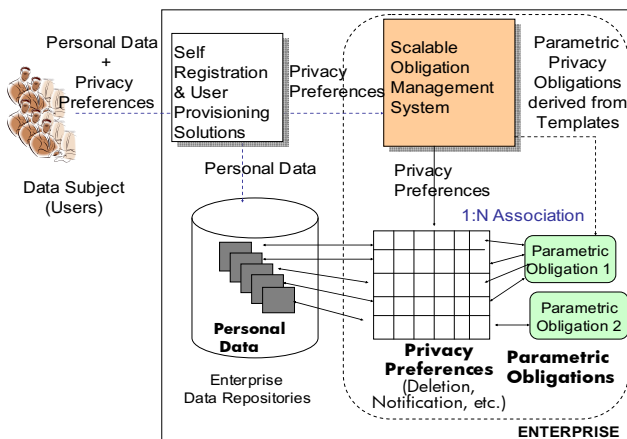
Fig. 15.11 Deployment of parametric obligation policies in organisations

Privacy administrators still need to interpret and refine privacy laws and guidelines and express them in the form of obligation policies that can be managed within their enterprise realities. Administrators must also understand how personal data is collected and where it is stored within an enterprise IT infrastructure.

They need to make decisions on which types of obligations an enterprise wants to enforce and which degree of customization (privacy preferences) to provide to their users (customers, employees, etc.). Once this information is known, obligation policies can be expressed in an explicit format, programmatically interpreted and enforced.

An administrator can leverage the obligation management framework and related scalable obligation management system – also referred in this work as SOMS system (see next section for more details) – to achieve this. In this context an administrator can use SOMS GUI capabilities to author parametric obligation policies by describing all their sections (Target, Events, Actions, On Violation actions, etc.). Via these GUI tools, for each parametric obligation, information is collected about which privacy preferences are required, how they relate to the policy and how to present this policy to the end user via a meaningful description.

A set of parametric obligation policies are then deployed in the SOMS system, to drive its privacy-aware information lifecycle management capabilities.

The SOMS system can be integrated with back-end Identity Management enterprise solutions, such as Self-Registration and Provisioning solutions [Cas04c]. In this context when users self-register (i.e. provide their personal information) via an enterprise portal, they are also presented with a list of supported (parametric) obligation policies along with the required parameters. Users can make their choices, select (a subset of) obligations and provide their preferences. These solutions provision users' information (personal data) in enterprise data repositories. Thanks to adaptors, they will also provision the SOMS system with the list of selected parametric obligation policies and preferences. The SOMS system, based on the Target definition of selected parametric obligation policies, knows where to store related preferences and ensure that links to personal data are maintained.

A potentially large set of users and their personal data ($> 100K$) can be provisioned to the enterprise. In this context, just a potentially small set of predefined parametric obligation policies is required to dictate all the criteria enabling privacy-aware information lifecycle management tasks. The SOMS system will manage them. There is no need anymore to instantiate an obligation policy for each provisioned data item: each predefined parametric obligation policy is dynamically associated to a set of managed data (that can change over time). This addresses the scalability requirement. The next section provides more insight on the Scalable Obligation Management System (SOMS) and its architecture.

15.3.2   System Architecture

Figure 15.12 shows a high level architecture of the SOMS system.
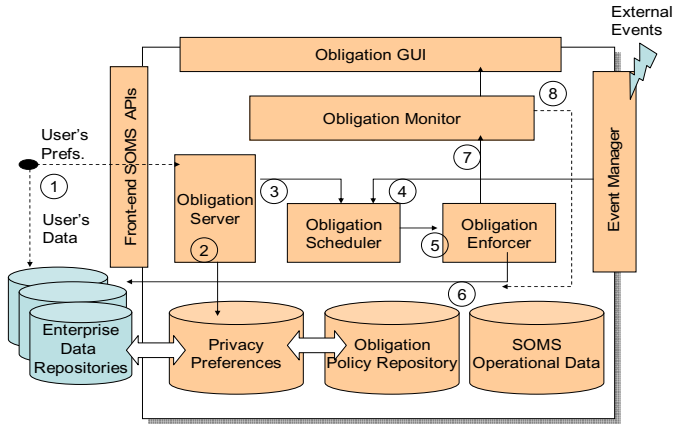
Fig. 15.12 High-level SOMS system architecture

The main components of the SOMS architecture are quite similar to the components described for the OMS architecture based on "non-parametric" privacy obligations:

- **Obligation GUI:** this is the graphical GUI used to: (a) author obligation policies; (b) check for their run-time status; (c) check the status of SOMS system components;
- **Obligation Server:** it is the core engine orchestrating interactions with other SOMS components. It interprets calls to SOMS APIs (1), stores privacy preferences in stated repositories (2), updates associations between preferences and parametric obligation policies. It provides information to the Obligation Scheduler (3) to ensure that the SOMS system is aware of the need to manage obligations on new personal data, based on specified preferences;
- **Obligation Scheduler:** it is the component that checks if (parametric) events trigger any parametric obligation (5). It solves, at runtime, any reference contained in the Events section of obligations (4), based on the contextual personal data;
- **Event Manager:** it is the component that checks for incoming external events (time, access, context events, etc.) of relevance of SOMS, translates

them into a meaningful internal format and transmits them to the Obligation Scheduler for further processing and correlation (4);

- **Obligation Enforcer:** it is the component that enforces the Actions part of triggered parametric obligations (6), by resolving, on the fly, related references, in the context of specific personal data and informs the Obligation Monitor (7);
- **Obligation Monitor:** it is the component that periodically checks the status of enforced obligation policies against the current status of data. Violations are reported and graphically visualized in the SOMS GUI. When specified in parametric obligations, this component will automatically try to remediate violations by executing the "On Violation" section of these policies (8).

The key innovation introduced in the SOMS system is its capability to dynamically interpret parametric obligation policies (i.e. their Target, Events, Actions and OnViolation Actions sections) and map their references on actual "targeted" data and preferences. This is done in an efficient way, via SQL queries that are instantiated on the fly – based on targeted data and related preferences.

Figure 15.13 provides a high-level view of the related process implemented in the SOMS system, triggered by the occurrence of external events of relevance for a given parametric obligation policy.
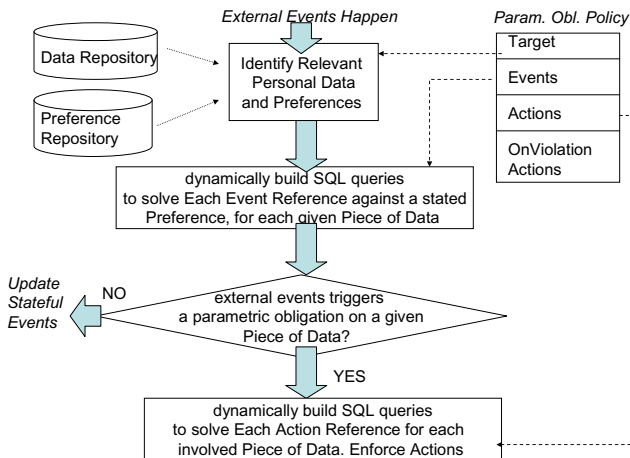
Fig. 15.13 Reference resolution process

When external events happen for a given parametric obligation, the SOMS system identifies the targeted personal data and related preferences. Based on this context, a few SQL queries are dynamically built to solve any reference in the Events section and, at the same time, check their values against stated Events conditions. For each piece of data (targeted by this parametric obligation) where the "customized" Events section triggers the enforcement of Actions, the system will dynamically build SQL queries to solve references in the Actions section and enforce them.

It is important to notice that some of the Events and Actions defined in parametric obligation policies might be stateful. For example, a parametric "Access Event", that is triggered once targeted pieces of data are accessed more than X times, has to keep access counters specific for each piece of targeted data. The status of enforced Actions has to be stored for monitoring purposes, etc.

The SOMS system stores all this metadata associated to parametric obligation policies in its internal "SOMS Operational Data" repository. Of course, this information can grow with the amount of managed personal data. However, it is just a matter of storage of simple data and efficient retrieval: this is done by using RDBMS databases and properly crafted queries (similar to the ones used to solve references). The SOMS system extends our previous version of obligation management system: it provides this new features in addition to the existing ones. As such the SOMS system can manage both parametric obligation policies and "traditional (non-parametric)" obligations. This allows an administrator to tune the system and take advantage of a hybrid obligation management approach depending on: (1) the need for efficiency and scalability (hence using parametric obligations); (2) the need for flexible and ad-hoc definition of obligations on specific instances of data (hence the usage of "non-parametric" obligations).

A full working prototype of our SOMS system has been implemented and re-integrated with HP OpenView Identity Management solutions, specifically its state-of-the-art User Account and Provisioning solutions for enterprises. This shows the feasibility of this approach in a real-world environment.

Initial results are very encouraging. Despite the fact that at this stage we cannot yet provide a quantitative analysis of SOMS performance, our prototype has been already tested with about 100K items of personal data - in a context where about 10 parametric obligation policies have been deployed (covering most common combination of event and action types). Each item of personal data was associated to specific privacy preferences. The SOMS system (installed in a "standard" PC using MS Windows XP Professional, with data stored in MySQL databases) has gone through all the required steps in terms of event processing, action enforcement and monitoring – without noticeable problems.

We are currently performing additional tests on larger datasets and different types of parametric obligations and collecting information on the behavior

of the system (future papers will provide this information). Future work includes further extensions of managed policies and performance tests.

15.4 Discussion and Conclusions

The management of privacy is very important for enterprises in order to deal with regulatory compliance and customer satisfaction aspects. In particular, privacy obligations need to be managed. In this context solutions are required to automate privacy-aware information lifecycle management and reduce costs. Their scalability to large data sets is a key requirement.

In this chapter we described work done in PRIME to explicitly represent, enforce and monitor obligation policies within organisations. A description of an Obligation Management System has been provided in full detail, demonstrating how to achieve this.

To address important scalability issues, this chapter has also discussed how to achieve this, by factoring in parametric obligation policies and a scalable obligation management system and framework.

A working prototype has been fully implemented and integrated with HP identity management solutions to show the feasibility of our approach in a real world domain. Initial tests demonstrate the scalability of our approach to handle obligation policies on large sets of data (more than 100K records).

16

Privacy Assurance Checking

Siani Pearson and Tariq Elahi

HP Labs

This chapter relates to the mechanisms developed within PRIME for privacy assurance checking: that is, for providing a greater degree of assurance to the user about the treatment of their personal data. This process involves checking that the services-side satisfies the requirements of the user with respect to the protection of their data, and then checking that the services-side really can and does comply with such policies. We also consider how to inject trust into the framework.

The following section provides an introduction to privacy assurance checking, giving an overview of the framework and the model, and how these fit into PRIME. The rest of the chapter contains sections covering more details about the system, comparison with related work, future plans for this work and conclusions.

16.1   Introduction

Trust is important to enable interactions on the Internet. People quite often have to trust e-commerce sites, service providers, online services and enterprises that they will perform as expected, provide agreed services and goods and will not exploit and misuse personal and confidential information.

The trust that people have in enterprises can be built, reinforced or modified via a variety of means and tools, including personal experience, analysis of prior history, recommendations, certification and auditing by known authorities. The behaviour of an enterprise, the fact that it will fulfil agreed tasks in due time and perform as predicted are all important aspects to shape its reputation and perception of trustworthiness. Related to this, the way an enterprise handles privacy aspects has also an important impact on trust.

An open issue to address is how to provide people with more customisable and fine-grained mechanisms to allow them to make judgments about the trustworthiness and privacy compliance of the remote receiver of their PII. For example, users might want to get some assurance of the capabilities of an enterprise, even before engaging in any interaction or transaction with this enterprise. This includes obtaining degrees of assurance that the enterprise can actually support specific privacy policies and obligations, that their data will be processed and managed securely, that enterprises' web services, applications and data repositories are installed, run and patched according to security standards and good IT practices, and/or that secure and trusted platforms are used. This section focuses on describing assurance policies that are used in order to provide a solution to this issue.

The problem of providing privacy assurance information has various aspects. The end user cannot be expected to be an expert on privacy matters and even less so on privacy technologies. An effective solution should cater to this and provide a means by which complexity is reduced and communication between the system and end user is unambiguous. Taking privacy seals as an example, the end user does not need to know anything about privacy except to trust that if a privacy seal is shown then the business can be trusted because it has been audited and verified by a trusted entity, like Trust-e. The lack of a privacy seal in itself can be an indicator that the business probably should not be trusted although it would be a stretch to say that the business would definitely treat PII in a bad way. Seals allow the end user to get a level of assurance about privacy without being experts in privacy technologies themselves.

It is part of the PRIME ethos that the end user should also be allowed to choose how their PII should be handled. To allow end user participation, unlike privacy seals which have no means of asking about the end user's choice, P3P, is an effort to give the end user some way of defining their own usage policies for their PII [Wor02]. P3P allows users to define their own privacy preferences for their PII which could be requested by on-line vendors during the course of transactions. If the business is also P3P-enabled then the user, or more conveniently a software agent, can compare the user's preferences against the business's and indicate the level to which the policies match and highlight the discrepancies. This brings participation from end users so that they feel more involved with the process.

Unfortunately, neither of the above provide any means to interrogate the business and its processes to see if the promises being made can be fulfilled [Pea06, Clab, Ack04]. The end user has to trust that whoever provides the seal or the privacy policy has made sure that the promises being made reflect the reality of the business's handling of PII. What is needed is for there to be some connection between what is stated on the privacy seal or P3P privacy policy and what really goes on within the business and its privacy capabilities [Pea06]. This brings us to the problem that was touched on initially which is that end users are not privacy experts. So any solution that involves

end users must maintain the discussion of privacy at a level that they can comprehend. This is tricky because privacy technology is generally complex and when deployed within already complex businesses processes it is more so. To make sense of this complexity would be beyond the scope of most end users' abilities. Instead of discussing privacy at this mind-boggling level, and according to PRIME principles, it is better to move the discussion to higher and more abstract levels where the business can express their privacy profile in terms that the end user can understand.

A related problem to the above is how much information to provide. The right amount of information should be sufficient for end users' needs and also not be too much of a burden for the business in terms of volume and exposure. Too much information would only overwhelm the end user and put a burden on the business's resources.

To summarize, we believe that a privacy assurance solution should communicate privacy preferences of end users to service providers, while also conveying service providers' privacy capabilities to end users in a universally understood language. In this chapter we discuss further what kind of assurance information to provide and how much, how high level privacy preferences and promises are mapped to to back-end capabilities and how we can provide trust in these mappings.

16.1.1   Scenarios Considered

The main driver for this work is that it increases user trust and willingness to engage in e-commerce and e-government. Example scenarios include:

- giving consumers the ability to determine whether unknown vendors on the Web are using IT systems and processes that can be trusted to execute their stated privacy policies
- automation of privacy assessment of the service side can be conveyed to the user in a more reliable and open way — for instance, compliance reports about enterprises could be accessible to the public, such as being available for viewing directly via a website — and with much more of a focus on evidence rather than having to rely on self-certification

Let us consider in more detail the example where a user engages for the first time with an enterprise that implements aspects of our model. In addition to other aspects that might be supported by the enterprise (such as seals and recommendations by other parties), users might require the enterprise to assure them about privacy practices, security and trustworthiness of their IT systems. Users might request the enterprise, by means of assurance policies, to provide them with fine-grained statements about their security systems and business practices and declarations of which privacy policies they support, specifically about how their data will be handled. The user could go even further by directly checking the trustworthiness of some platforms, via TCG-enabled mechanisms [Tru03, Tru06] if supported. The user can use their

compliance checking system to verify enterprise statements and promises, re-member their expectations and re-check them over time. If the user is satisfied by these initial statements, they might decide to engage in an interaction or transaction with the enterprise and potentially disclose their personal data.

16.1.2   How Assurance Checking Fits in with the PRIME Approach

The motivation for developing assurance control fits in with the strategic objective of the PRIME project, which as discussed earlier in this book, is to research and develop approaches and solutions for privacy-enhancing identity management that can empower European citizens to exercise their privacy rights, and thus enable them to gain trust and confidence in the Information Society. In order to increase user control over the release of their personal information, we wish to provide a mechanism to help the user to assess the trustworthiness of back-end systems before releasing such information. This mechanism can also be used in order to allow the user to check the proof of properties contributing to trust (such as the validity of seals of approval), and not have to rely upon assertions by companies that could be deliberately or erroneously false or misleading. In addition, it can be used to help 'good willing' enterprises — that are aware of the importance of trust as a driving factor to underpin privacy and the importance of privacy for reputation and a business enabler — to ensure that their trust and security are operating as expected and to comply with legislation. The most basic starting point for building trust is just to check that the services-side has a PRIME system that is operating correctly and via education enhance the user's trust in the body certifying this system.

We recognise that the problem cannot be solved by deploying technologies alone: behaviour and implementation of correct process are very relevant. However, our objective is to build technical solutions that can help enterprises increase automation and give people additional support in making informed decisions about trust.

In the PRIME model, end-users formulate their (individual) privacy pref-erences before interacting with an organisation, and can negotiate the proofs that need to be provided to an organisation. In some situations, zero knowl-edge techniques could be used and the end-user could remain anonymous, although in other situations PII may need to be transferred in order for the particular type of transaction to go ahead. This negotiation is automated, al-though input may be given by the user in complex situations. Following W3C recommendations [Wor02], in order to encourage adoption of this approach by service-providers, it would probably be necessary to have the service-side start the negotiation process by transmitting an initial set of requirements and options to the end-user.

Several different approaches and techniques are possible:

- *Anonymous checking of service side:* End-users could check up-front the fulfilment of specified back-end (enterprise) properties or trust requirements (for example, whether the service side could support obligations or was providing a secure processing environment) before deciding whether or not to proceed with a transaction.
- *Negotiation of 'sticky' policies:* End-users could be offered a choice of trust requirements by the service provider which would then be customisable; alternatively, end-users could add new trust requirements into the negotiation process (between the end-user and service side). The resultant negotiated policies can 'stick' to personal data and as it moves around the back-end these policies will be enforced; these policies can include trust requirements.
- *Compliance checking by 'good willing' enterprises:* The service side can automate checking of trust and assurance necessary conditions in access control policies.

This model where end-users formulate their (individual) privacy policies before interacting with an organisation, and then have the organisation verify that it will comply with the end-user policy, is in contrast with much current practice where at best an end-user looks at an organisation's policy and decides if it is acceptable. Thereby, this model is supporting users to maintain control over their personal spheres and thus to technically enforce informational self-determination, and hence is in accordance with the philosophy and motivation of PRIME. Informational self-determination is a core aspect of privacy and is in many countries acknowledged as a basic human or constitutional right. As an end-user it would be preferable to have the possibility (in case the user so desired) to dictate or customise some of the privacy policies, rather than passively accept whatever is dictated by the enterprise [Kob02, Kob03]. The desire for this is supported by various studies that highlight end-users' concerns about privacy violations. For instance, according to a study by the UK Information Commissioner [Com03], 40% of the UK population are classified as "the Concerned" who have proactively protected PII through withholding it. They are less likely to purchase products if they have to give away too much information, and their attitudes towards organisations are likely to be influenced by a reputation for good information handling practices. This study also classifies 13% of the UK population as "the Proactive", who prefer working with companies that excel at good personal information management.

Based on results of a meta-analysis of user surveys related to Internet privacy in Europe, social research within the PRIME project has derived the importance of trust assurance methods for (re)establishing trust in online relationships as an important social requirement for PRIME technologies. Besides this, the meta-analysis also indicates a preference of many users to have more transparency and better user control over the use of their online behavioural data. Also, usability tests conducted on PRIME early prototypes

and user interface mock-ups of identity management systems showed that many users distrusted the tested systems and were also pointing out the users' need for trust assurance methods. See Chapter 20 for more background and discussion about the HCI research within PRIME.

Turner has carried out various studies to assess factors that affect the perception of security and privacy of e-commerce web sites [TZY01, Tur02, Tur03]. He concludes that consumers depended on recommendations from independent third parties to ensure security [Tur03]. This supports our view that end users would find it helpful to be able to be given enterprise assurance details provided by third parties, such as privacy seals.

Note that it not necessary for people to have to author policies, because default policies can be provided which they could use, preferably vouched for by entities that they trust (for example, consumer organisations).

16.1.3 Assurance Control Framework: Overview

A compliance checker component has been developed as part of the PRIME architecture, which is used both on the services-side and on the user-side (as there is a mirrored design).

Within the PRIME architecture, the Assurance Control component is in charge of handling assurance policies that cannot be directly managed by the Identity Control (IC) and Access Control (AC) components. It provides assurance and trust based on policies that are created by users and service providers, as discussed already in Chapter 13.

Within PRIME this component is implemented as being separate from the Access Control Decision Function (ACDF) component, but it could in fact be implemented as part of ACDF and part of other components such as Identity Control (IC) (for checking trust constraints when access control is not invoked). An example of the latter case would be if someone wished to check the trustworthiness of a service-provider upfront in a 'preamble' phase before provision of any identity information (potentially in a fully anonymous manner).

Our focus of is on the user-side and its interaction with the services-side. The emphasis is on ease of use, practicality, and unobtrusiveness of the assurance control component from the users' perspective and flexibility, granularity, and ease of deployment from the service side's. Privacy policies form the basis of communication between client and service. Since, as described in Chapter 13, policies are created using privacy statements written in natural language, it is hoped that both clients and services will be able to understand each other more clearly. This also empowers a client, who it is assumed has no technically advanced knowledge, to communicate their privacy preferences in a language they understand. Assurance Control provides the tool set to create, modify, and select privacy policies according to the clients' requirements, as well as allowing the service provider the same tool-set to craft their own policies. To reconcile the two parties' policies, which are probably different,

the Assurance Control provides a policy matching engine to help the client identify the deficiencies and strengths of the service provider's privacy policy.

To verify the validity of the privacy policy Assurance Control provides the client a means of checking the whole policy or just selected clauses. The Assurance Control handles the requesting, gathering, and output of test results for the client's review. If the client is satisfied they accept the service provider's policy.

16.2   Privacy Compliance Checking System

In this section we provide more details about the privacy compliance checking mechanisms that were developed within the PRIME system.

16.2.1   Design Rationale

In order to provide users with greater choice and control, we believe it is beneficial to offer users the option to check the degree of evidence that the service provider can provide that they can be trusted to process the user's PII in a privacy-friendly manner. To this end, we have implemented an "Assurance Control" (AssCtrl) component within the PRIME framework. The main functionalities provided by this component are to:

- Compare the service provider's privacy policies with the user's privacy preferences and highlight similarities, differences and deficiencies.
- Conduct capability tests to verify the statements made in the service policy and to ensure that the service side is capable of fulfilling the promises made in their policy.
- Provide results of the above in a way that allows a user to make informed decisions about releasing their personal information when this is necessary for delivery of the service, with some guidance built in.

16.2.2   Architecture

This section gives an overview of our system within PRIME that handles assurance policies in enterprises. Figure 16.1 shows the core aspects of this model.

The model supports the following core interactions between users and an enterprise:

- Users ask an enterprise to demonstrate their support and compliance to a set of policies (cf. 1 of Figure 16.1): this can be done by users before engaging in any interaction with the enterprise. The "policy compliance checker" module, within the enterprise, issues compliance statements and potentially it supports degrees of verifications made by users. For example, users could require that the enterprise will protect their data to a
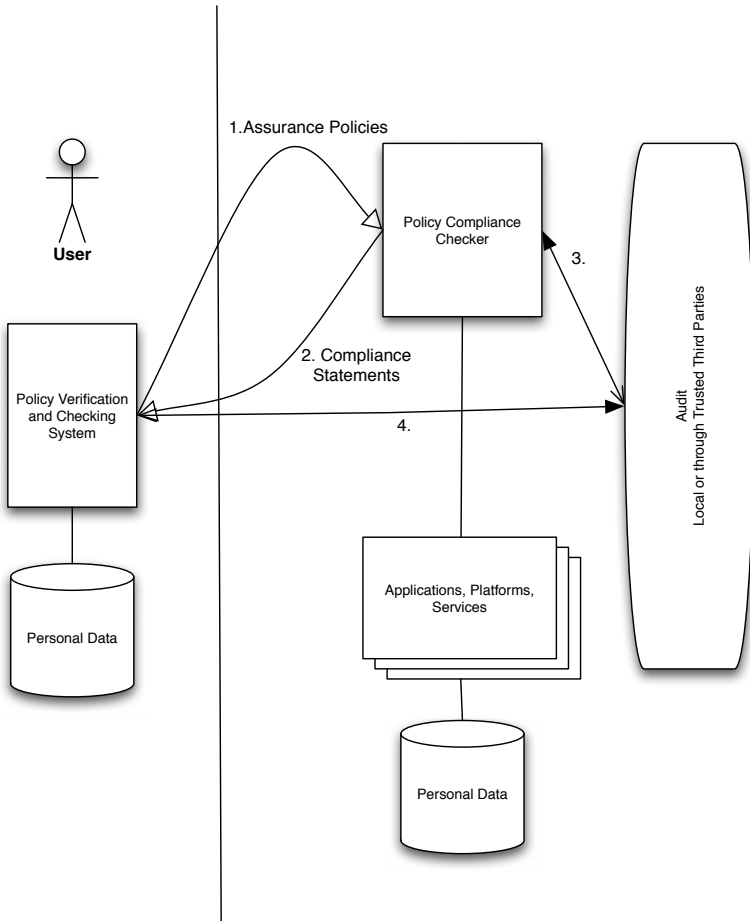
Fig. 16.1 High-level architecture of a policy compliance checker

specified level of tamper resistance, that the enterprise is not running certain software that has known bugs without the requisite patches, that the enterprise can support obligation checking or that the enterprise has a certain type of privacy seal. The outcome is recorded and remembered by the "policy verification and checking system" on the user-side for future reference and control. A similar mechanism can be deployed in enterprises in federated contexts where the enterprise needs to disclose data subjects' personal data to other parties (during business interactions and transactions).

- Users disclose their personal data along with their privacy obligations (cf. 4 of Figure 16.1): users can dictate the set of privacy obligations and constraints they want to be fulfilled on their personal data.
- Users control and verify their expectations and compliance over time (cf. 2,3 and 4 of Figure 16.1): the 'policy verification and checking system', at the user-side, remembers commitments, obligations and promises made by an enterprise. It processes them against evidence and information provided by the enterprise and potential third parties in order to verify their consistency and compliance. This module provides users with intuitive visual clues that help them to make decisions and influence their perceptions of the trustworthiness of an enterprise in executing what has been agreed.

The policy compliance checker we describe has a privacy focus. The compliance checking may include consideration of the usage, configuration and availability of organisational resources such as database systems, firewalls, hosts, virus scanners and privacy seals, as well as system and application properties, such as host patching or Trusted Platform Module (TPM) self test, together with user provisioning and maintenance and checks that IT controls are working as expected.

16.2.2.1 Overview of the Architecture of the Assurance Control Component

The PRIME Assurance Control component is a specific implementation of the generic system shown in Figure 16.1.

The following internal modules are present in the component and are depicted in Figure 16.2, which shows their relation to each other:

1. **Policy Handler:** This module is the only interface between the Assurance Control and the rest of PRIME. It provides access to public Assurance Control methods (cf. Policy Interpreter and Handler, Figure 16.1).
2. **Trust Aggregator:** This module is responsible for collecting results from various platforms and conducting certain system level tests. At present it is limited to checking for facts about back-end systems. It stores its results in the Results Database.
3. **Policy Compliance Orchestrator:** This module is responsible for calling the Trust Aggregator, compiling the results and then using rules to ascertain the level of assurance of the back-end. It can also check assurance based on passed assurance information by way of ontologies and the Reasoner component.
4. **Matching Engine:** This module is responsible for parsing the client and service policies and comparing them. It highlights the differences, similarities and deficiencies in the service policy as compared to the client policy. Chapter 13 shows the data format of the policies and an example.
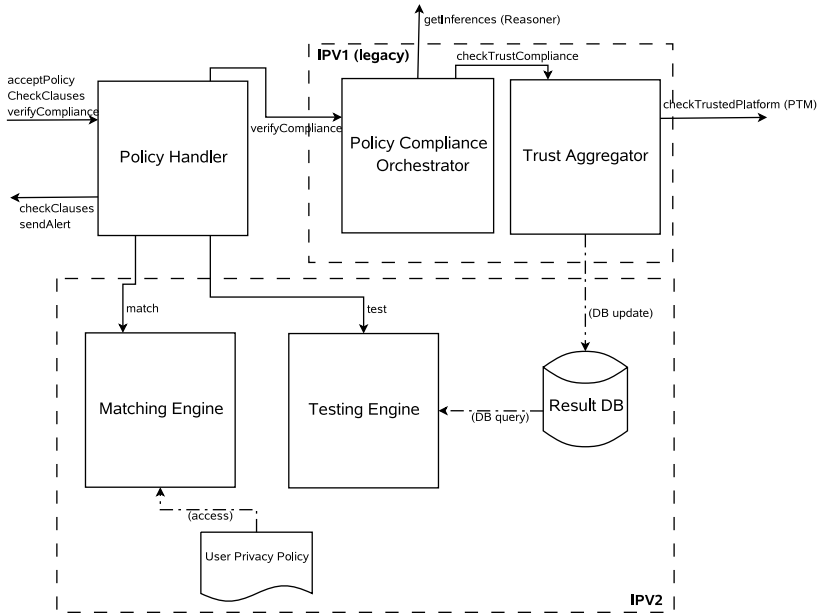
Fig. 16.2 External and internal API view

5. **Testing Engine:** This module takes the list of clauses passed to it, finds the corresponding test results using the mapping table, and then retrieves those results from the results database.

Additional functionality shown in Figure 16.1 is contained in additional databases or distributed throughout the PRIME system.

16.2.2.2   Feature Overview

- **Policy Matching:** Allows the user to compare their prefered privacy policy against the service provider, given a single context, and then decide if the service side is providing sufficient coverage of their privacy concerns. The user will have some guidance via templates which will indicate which clauses ought to be present in a sufficient privacy policy. This feature is critical on the client side, and should be included in IPv3.
- **Checking/Verifying privacy policy:** The client will have the ability to gather assurance information from the service provider in order to determine if the back-end has the capabilities of carrying out the stated policy. This feature is critical on the client side and impacts the service side, and should be included in IPv3.
- **Feedback:** Preferably, the client should have the option to provide feedback at all junctures if they wish to communicate their complaints or

suggestions to the service provider, in the context of privacy policies. This will be useful for service providers in order to gauge if their policies are adequate and if customers are happy to accept them. This feature is moderate level on the client side and not required on the service side.

- **Provisioning Assurance Information:** Various systems checks are implemented in order to populate the result database. From this basic assurance information more complex checks can be done where the simple checks are grouped together. This feature is critical on the service side. See [PA09] for further details about how this process works.
- **Trusted Third Party Verification:** The trust providers will be selected initially from a well known and popular group. The user will have the ability to select the ones they would like to use, or even add new ones for themselves. This feature is moderate on the client and services-side. This will allow end-users, or their agent, to check the validity of the privacy policy they receive from the service provider. This is a medium-level feeature. We highlight these options via the client user interface.

16.2.3   Key Interfaces

The key interfaces of the Assurance Control component, supporting the above functionalities, are:

- *acceptPolicy*: the main function that triggers the assurance control process; it incorporates both matching policies and verifying clauses. It is called by the server when it wants the client to agree to a privacy policy. It invokes the `MatchingEngineImpl` class to do the matching of the users preferences against the service providers privacy policy. It might be that the requesting component is on the client side, such as the Access Control module.
- *checkClause*: this function provides a means to verify the validity of clauses without the matching phase. It is called by the client, locally or remotely, to retrieve the results of the clauses that were to be checked in the clauselist file. It is usually called after the client has matched the service and client side privacy policies, but it can be invoked at any time as long as a clauselist file is supplied.
- *verifyCompliance*: this function checks whether the operational system really does satisfy the clauses claimed by the service side.

Figure 16.3 shows how the assurance control mechanisms works, at a high level.

The *checkClauses* interface requires that a remote Assurance Control module be invoked to conduct tests and return the results of the tests. To facilitate this transaction there must be some way to invoke the remote module and receive its results. The Identity Control component (IDCTRL) is seen as best suited to facilitate this protocol and is relied upon to handle the transmission of data and the invocation of remote modules.

Pre-conditions: The module assumes that privacy policies have been created prior to invoking any of the functionality. It also presupposes that universal identifiers are used for statements so that all interacting parties have common knowledge of what is being referred to by each statement present in the policies. One last assumption is that inter-Assurance Control communication will be synchronous.

Post-conditions: The module does not present its results to the end user and must rely on an intermediary, such as the Console, to display the results in a way that is pleasing to the user and makes logical sense.

As seen in Figure 16.3, the Assurance Control component is invoked whenever another component requires some assurance information. It is separate from the access control functionality (ACDF) hence allowing it to be utilized by other components independent of access control. The Identity Control component (IC) is a central protocol driver in the PRIME implementation and as such handles communication and protocols for the other components present. The Identity Control mediates the interaction with the requester, and signs any statement provided by the Assurance Control component (potentially involving signature via a Trusted Platform Module (TPM) — the hardware security chip in a trusted platform). The disclosure of assurance-related information is finally subject to a decision by Access Control.
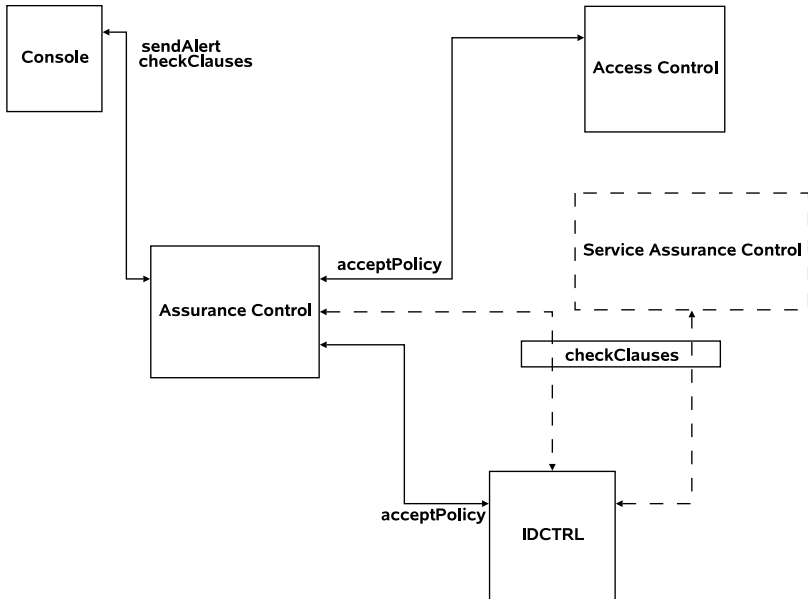
Fig. 16.3 Interactions between PRIME components and Assurance Control

16.2.3.1   Details

From IPv2 specification onwards of the Assurance Control component there have been refinements and enhancements made in the way trust and assurance in the service side's privacy policy is achieved. The key refinements and side effects are:

- Assurance is being abstracted into predefined statements about privacy and hiding the details of topology, infrastructure and other intricacies from the user.
- The user will be able to develop their own privacy policies using a pool of predefined privacy enhancing statements and collect them into a policy. There can be many policies depending on the context of the transactions. This is done off-line using the provided Assurance Policy editor prior to utilization.
- The service provider will use the same pool of predefined privacy enhancing statements to construct policies that are cognizant of their internal controls and privacy regulations. There can be many such policies depending on the context.
- The service provider will put into place
    - the proper privacy enhancing mechanisms into their infrastructure so that each statement in the policy is covered and
    - have these control mechanisms validated by a third party and obtain a token of proof for presentation to the user that this validation has taken place.
- It is not the user who is responsible for validating the suitability or appropriateness of the privacy enhancing infrastructure of the service provider, but a trusted third party. The user will only be responsible for checking that third party seals are current and valid and accessing the trustworthiness of the vouching party.
- Trust has been moved from the certificate provision service (which converts certificates to the required format and adds in privacy-related meta-level information) to appropriate third parties that validate the deployment of privacy enhancing mechanisms.

The problem of providing assurance has been split into two distinct activities and realms.

1. The first is the creation of privacy policies. Both users and service providers will have the freedom to create policies to suit their needs. In order to bring the two together a common vocabulary is required. This comes in the form of privacy statements or privacy clauses. This is one of the two critical aspects of the assurance scheme. The clauses will be created carefully and worded in such a way to be privacy positive: that is to say that a clause will never reduce the level of privacy afforded to the individual. The clauses will not be concerned with technical implementation

details, only statements about privacy as required by law, good business practice, and consumer protection will be present. This abstracts away the technical details from the essence of the statements which are only concerned about what should happen with PII and not how it should happen. This prevents restrictions on the way the solutions are implemented and also prevents users from having to be technically savvy to use this scheme. To protect users from having to understand technical details about privacy products and construct detailed policies which may be removed from practical reality, users are only required to select those statements, or clauses, that they think they require a service provider to adhere to. In the same vein a service provider, although more technically knowledgeable, can speak the same language as its users and can communicate its responsibilities clearly. See Chapter 13 for more details.

2. The second critical aspect is once a policy has been set by a service provider the onus is upon them to implement the measures to uphold those policies. The fact that clauses only talk about the "what" and not the "how" allows the implementation to be carried out befitting the service provider. To ensure that the mechanisms put in place are appropriate a third party needs to be involved. The exact relationship of the third party to service providers and users is not a factor as long as users trust their evaluations. The Assurance Control provides assurance information, results of verification tests, and validations of third party approvals. The user only has to decide about the trustworthiness of the third party. See Section 16.2.7 for more details.

The Assurance Control may have a role in handling policies on client, server and third party sites. Within PRIME, we are most interested in the cases where the back-end system that handles PII is to be evaluated, either locally or remotely.

16.2.3.2   Data Formats of Mappings

The mapping of the privacy policy to system level checks is represented using an XML format. The DTD for that is:

```
<!ELEMENT clause ( tid+ ) >
<!ATTLIST clause gid NMTOKEN #REQUIRED >
<!ELEMENT gtidmap ( clause+ ) >
<!ELEMENT tid ( #PCDATA ) >
```

An example of a mapping is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE gtidmap SYSTEM "gtid.dtd">
<gtidmap>
<clause gid="1">
<tid>1</tid>
<tid>3</tid>
```

```
</clause>
<clause gid="2">
<tid>2</tid>
</clause>
<clause gid="3">
<tid>1</tid>
<tid>2</tid>
<tid>3</tid>
<tid>5</tid>
</clause>
</gtidmap>
```

Where `tid` is the test id as defined by the service provider and which links to a certain test result in the Results Database.

The list sent to the server for clause checks follows the following DTD:

```
<!ELEMENT cclist ( ctid, clause+ ) >
<!ELEMENT clause EMPTY >
<!ATTLIST clause gid NMTOKEN #REQUIRED >
<!ELEMENT ctid ( #PCDATA ) >
```

An example clause list is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cclist SYSTEM "cclist.dtd">
<cclist>
<ctid>1337</ctid>
<clause gid="1"></clause>
<clause gid="2"></clause>
<clause gid="3"></clause>
</cclist>
```

16.2.4   Implementation Details

16.2.4.1   Comparing Service Side and End User Assurance Policies

A description of the structure of service side and end user assurance policies has already been given in Chapter 13.

Policy matching occurs on the user's side, as shown in Figure 16.4. The arrows indicate the direction of clause flow, first to create policies, and then to communicate privacy preferences. The authority entity maintains the pool of clauses and templates, distributing across the internet to make it easier to locate them. The integrity and authenticity of the clause pool and templates would need to be maintained by using signing techniques in use today (such as MD5), or some other mechanism.

During a transaction where PII is to be divulged to the service provider, the end user can conduct a policy matching activity where the system can compare their privacy preferences (as stated in their privacy policy) to that
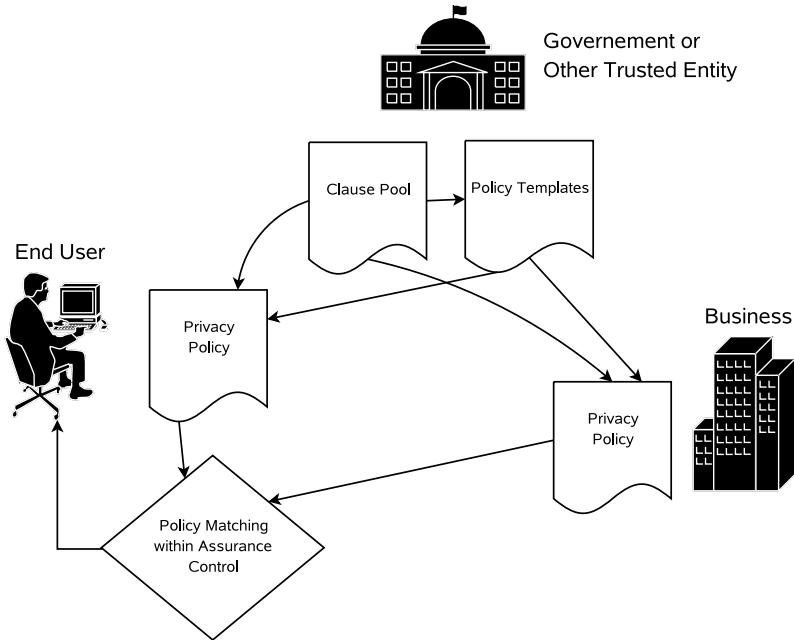
Fig. 16.4 Privacy policy creation and matching

of the service provider's policy. While comparing, or matching, it must be remembered that both the user and service provider are drawing from the same pool of clauses. So it is simply a matter of checking if the end user's clauses appear in the service provider's policy.

16.2.4.2   Clause Mismatches

The trivial case is when both the end user and service provider policies are identical. In this case there would be no warnings. When this is not the case then the system has two scenarios:

- **Missing Clauses:** This occurs when the service provider does not provide a clause(s) present in the end user's policy. This is flagged by the system and reported to the end user.
- **Excess Clauses:** This occurs when the service provider's policy has a clause(s) not present in the end user's policy. This is not a cause for alarm since all clauses are privacy positive and the additional clause will only strengthen the privacy policy.

After the matching phase the end user can make a decision on whether or not to divulge their PII. The results of the matching phase are only there to help the end user make an informed decision about their privacy concerns rather than underlying privacy technologies. They need only make decisions

about missing clauses and not worry about the underlying details. They can take the results of the matching phase at face value or then move to the next stage of the process which is validation of the clauses against capabilities of service provider's backend systems. We talk about this in Section 16.2.5.

It is useful for end users and service providers to give and receive feedback of their concerns and experiences. We have enabled this by allowing a user to give feedback at the time of abandoning the transaction. For the service provider, a clause represents a translation layer which is easy to understand by end users and hides the complexity of the underlying technologies. It also decouples the implementation of clauses from their expression and hence allows the service provider flexibility in implementing their back-end solutions to best fit their business processes and needs.

The standardization of clauses and provision of templates to ease the creation of policies addresses the points we made about establishing a common privacy language between end users and service providers. This discussion has also established that the level and amount of assurance information needed by end users can be curtailed by abstracting privacy concerns from implementation details. We now move on to discussing the relationship between the privacy clauses and the actual privacy capabilities of the service provider.

16.2.5   Mapping and Capability Validation

Once a policy has been set by a service provider the onus is upon them to implement the measures to uphold those policies. The fact that clauses only talk about the "what" and not the "how" allows service providers flexibility in choosing the best solution for their particular infrastructure. To tie together and bridge the "what" to the "how" there has to be some sort of mapping that facilitates this connection. The main job of this mapping is to communicate the back-end privacy controls, processes, and other privacy enhancing features implemented by the service provider through the process of verification of clauses in privacy policies.

The service provider will ensure that only those clauses are present in their policy that can be upheld in reality by the back-end. The problem is now to have some way to translate a back-end control, which is ignorant of our solution, to something that can be stated in terms of clauses. Our solution allows each clause to be composed of specific tests that query controls and system components on the back-end. In this way a suite of tests can be created that inspects the system and reports back the results that can be used to verify clauses. Figure 16.5 shows how each clause in the privacy policy is mapped to back-end tests. A test only validates that the control or feature is in place and working in a known manner. There can be multiple tests on the same control to validate particular attributes, as long as they are relevant to the clause being verified.

For example, for the clause: "All data in storage will be secure against unauthorized access" the corresponding tests could be:
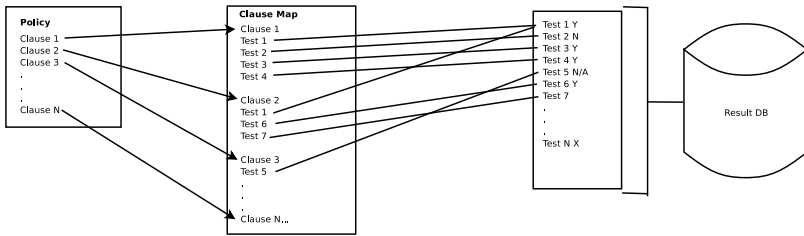
Fig. 16.5 Mapping clauses to back-end controls through tests

- Test 1: Check hard drives are encrypted
- Test 2: Check access control subsystem is functional
- Test 3: Check that no unauthorized accesses have occurred

The labelling of the tests is arbitrary; as long as a unique identifier is assigned and the mapping table is correct and updated then clauses can be mapped to any group of tests. In our solution we have opted to store the results of these tests in a result database, and updating it everyday in a batch process. There is nothing preventing realtime testing of the system, as long as performance issues are taken into consideration.

An advantage of keeping a suite of tests where each test can be used by multiple clauses is that this could help reduce the management and upkeep of the test suite and help reduce the overhead of utilizing our solution. It should be noted that we are assuming that the relationship between tests and controls has been established by the service provider or someone who has the expertise.

Once the test suite has been created the proper mapping between clauses and back-end controls, via tests, has been established the service provider can now offer the end user a way to verify the claims made on the service provider's privacy policy. This step, called capability checking, is crucial in affording assurance to the end user since it allows the user to see if the service provider is actually able to uphold their promises.

A simple walk through is shown in Figure 16.6, corresponding to the following steps:

1. The user, having selecting which clauses they want verified, submits these to his or her capability checking, aka Policy Validator, module.
2. This module communicates this list to its counterpart on the service side and awaits its response.
3. The service-side Policy Validator searches for the clause to test mapping in the mapping file kept on the service side. It then queries the result database for these tests and retrieves their results. It can either aggregate the test results to a level that only verifies that the clause was fulfilled or it can send back more information. This is configurable and left up to
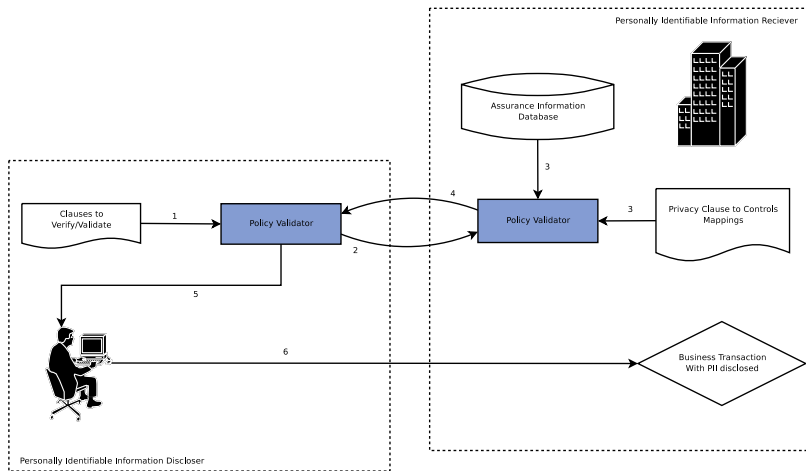
Fig. 16.6 General protocol flow

service providers to choose how much detail they want to include in test
result data.

4. The results are transmitted back to the awaiting client side.
5. The Policy Validator displays the results to the end user to allow them to
make an informed decision about releasing their PII.
6. If the end user is satisfied then they can divulge their PII or if not they
can provide feedback to the service provider so that it can make meet user
demands in the future.

So far there has been implicit faith placed in the service provider to do
the right thing. We have assumed that the correct back-end controls are in
place to ensure the privacy of end users' PII and only those clauses have been
put into the privacy policy that are backed up by those controls. This is an
obvious area of abuse and so trust has to be introduced here. In our solution
trust comes in the form of third parties. We shall see in Section 16.2.7 how
they interact with service providers and end users and how they fit into the
solution being developed here.

16.2.6   Description of Protocol

This subsection looks in more detail at the interactions of the user and service
side Assurance Control components with other PRIME components during a
complete protocol run. The protocol runs as an optional loop within the larger
PRIME protocol, before the requested information is revealed by the user-side
to the services-side. The user has the option of skipping this step by choosing
not to perform the assurance checking stage on their console when the service
side policy is presented to them. The implication of such a choice will be that

there will appear in the audit logs an entry stating that the assurance checking stage was skipped. If the user has initiated a check then they will have the capability to review the assurance policy, conduct and view results of tests and finally to accept the assurance policies of the service provider or to break off from the transaction.

We are also assuming that the user has already set up their assurance preferences off-line before initiating the protocol. The first time the user visits a site and has to provide PII, the Assurance Control will be invoked. For subsequent times the Assurance Control will only be invoked if sufficient change has occurred in the assurance characteristics of the service provider. The IC is the main invoker of the Assurance Control and thus other components wishing to interact or utilize the Assurance Control should channel their requests through the IC, this is due to the IC being capable of session management, context handling, addressing.

The IC invokes the Assurance Control when it sees that the client has not previously accessed the site or that the server side does not hold any assurance policy information for the client. It is important to note that the Assurance Control can be invoked by the IC for any useful purpose and is independent of the AC, but in this example we are assuming that the AC has been invoked and it is waiting for an affirmative response from the Assurance Control of the assurance checking step.

16.2.6.1   Two Alternatives for the SPCC Flow

As an alternative to the user choosing to make the assurance check, the assurance check could instead be performed as a background process and the result summarized in the "Send Data?" dialog. This has the benefit of giving a simplified approach. For this approach, there must be some means of letting the user find out what privacy functionality checking is all about (e.g. by clicking to be shown help text), seeing a breakdown of the results of the tests (by clicking an overall assessment icon to allow another expanded screen) and viewing/customising which policies are checked and which third parties are involved. These approaches are discussed in more detail in Section 20.

An explicit choice might be preferred if some checks take a long time to perform which would make automatic (and mandatory) checking an obstacle. It may also be preferred for transparency and enhancing user choice, although the approach above would probably be simpler.

Since efficiency is an issue within PRIME, we have it that the user explicitly makes the choice to run a SPCC check.

16.2.6.2   Example Flow

An example end-to-end protocol flow will be the following: an optional assurance management loop where the user requests assurance from the service side (SS) and the SS returns assurance information. Preconditions are that the

client has not previously accessed the site (although this can be overridden), that the user has set up assurance preferences previously, and similarly the service provider has predefined policy entries, that the third party has carried out testing (although if desired this could occur dynamically) and that the results of this are stored in a database accessible from the service provider side. We also presume that the user chooses to make the assurance check.

This protocol flow could happen at different stages in the user-SS interaction, but typically after the claim request and Data Handling Policy (DHP) is sent from the SS to the user and before the claim and evidence and DHP is sent from the user to the SS.

An example protocol flow is shown below. If desired, this could be further simplified, for example by combining the information in the two windows displayed to the user into one, and possibly giving less options to the user for control and just the final assurance information (or information about the lack of it).

The protocol below runs as a step within a larger PRIME protocol, before the user actually sends the information needed for the service provision to the service side. The first time the user visits a site and has to provide PII, the Assurance Control will be invoked. For subsequent times the Assurance Control will only be invoked if sufficient change has occurred in the assurance characteristics of the service provider.

Note that if there is a mismatch in the service side and client policies then user intervention is required. Also, the user's choices on the console will drive how the IC behaves.

An example break down of the protocol follows beginning from the SS.IC:

1. The SS.IC sends the appropriate service assurance policy (SS.AP) to the US.IC. (This could alternatively have been done when the data policies are transferred to the client.)
2. The US.IC passes along the AP to the US.Assurance Control.
3. The US.Assurance Control compares the SS.AP with the US.AP (which is retrieved according to the context of the session, e.g. financial, government, enterprise etc.)
   a) If there is a perfect match we go to stage 5.
   b) If there is a mismatch we goto stage 4.
4. US.Assurance Control sends an alert detailing the mismatches to the US.IC which passes it along to the US.CONSOLE. A window is displayed to the user.
   a) If the user is satisfied that the mismatch is nothing to worry about then goto stage 5.
   b) If the user is unhappy and wishes to discontinue the transaction end the protocol and allow user to give feedback through the US.CONSOLE.
      i. If feedback is given send it to the US.IC, which sends it to SS.IC at which point it is stored in the SS.FEEDBACK storage area for

       review. Send terminate signal to US.IC which negotiates with the SS.IC and breaks the connection.

    ii. If no feedback is given then send terminate signal to US.IC which negotiates with the SS.IC and breaks the connection.

5. Send the SS.AP to the US.CONSOLE for review and allow the user to select clauses to verify system capability at the SS.
   a) If no clauses are selected then goto stage 13.
   b) If clauses are selected then goto stage 6.

6. Send the selections from the US.CONSOLE to the US.IC which passes that data to the US.Assurance Control which creates a shorter version (US.APCHECK) of the full SS.AP with only those clauses to be checked.

7. The US.Assurance Control sends US.APCHECK to SS.IC which passes it on to the SS.Assurance Control.

8. The SS.Assurance Control takes the clauses present in the US.APCHECK and conducts the tests relevant to those clauses.

9. The SS.Assurance Control gathers the results of the tests and creates a SS.APRESULTS file similar to the US.APCHECK with the clauses it tested and the results appended for each test with any verification signatures also present.

10. The SS.Assurance Control sends SS.APRESULTS back to SS.IC which passes it to the US.Assurance Control via US.IC.

11. Once it receives SS.APRESULTS the US.Assurance Control verifies that the clauses present are the same as those in US.APCHECK.
    a) If they are then it passes the results to the US.IC for display on the US.CONSOLE. It also records the results of the tests in the US.AP
    b) If they aren't the US.Assurance Control goes back to stage 7. It keeps track of how many times it has done this in one session and tries three times.

12. A window is displayed to the user. The user can now decide from the results displayed on what to do next.
    a) If they wish to check more clauses the protocol goes to stage 6.
    b) If they are satisfied with everything the user accepts the SS.AP.

13. The US.CONSOLE sends the accepted policy back to the US.IC which passes it to the US.Assurance Control.

14. The US.Assurance Control stores the SS.AP with a reference to the transaction as well as the results of those clauses that were checked by the user and all the data relating to those checks.

15. The US.Assurance Control sends the US.IC that it has no objections to release of PII.

16. The US.IC sends notification to the SS.IC that the US accepts the assurance policies of the SS. It also sends a positive response to the US.AC that it can go ahead and release the PII.

17. The SS.IC sends a message to the SS.Assurance Control to make a record of this transaction with reference to the SS.AP for future use.

18. The US.IC and the SS.IC are free to carry out their normal functions uninterrupted by either Assurance Control.

The preliminary result is that assurance results are returned to the user (within a User Interface (UI) that indicates whether assurance policies on the service side match user preferences and if so, whether they are currently valid). The user then uses this information to help him/her decide whether to proceed or whether to abandon the transaction with the option of giving feedback to the service side.

If the flow continues, the user makes a release decision, in other words that the user may abandon the transaction, or else continue on with the claim and evidence and the DHP being sent from the user to the service side. The result is that the claim (cf. service side policy) and associated assurance evidence is sent from the service side to the user. This may be followed by verification, access control, and logging.

16.2.7 Role of Third Parties within the Trust Chain

The service provider can put a great deal of effort into developing their back-end to be privacy enhancing and mapping the clauses in their privacy policies to their controls properly, yet still not achieve trust in the hearts and minds of end users. The missing trust has to come from entities that end users do trust such as trusted third parties (TTPs), like Trust-e and Verisign[ver], or non-government consumer organisations. The way forward is to invite the TTP to scrutinize their back-end systems, the mappings and their privacy policies. If the TTP is satisfied it would issue a trust token that can be presented to the end user at the time of policy matching and verification, thus providing trust in the results and ultimately in the business.

The main concerns of the TTP are:

- Verifying that the controls and privacy enhancing technologies that are implemented by the service provider on their infrastructure are configured and functioning properly
- Verifying that the tests used to interrogate the proper configuration and function of are capturing and analysing the correct data
- Verifying that the clause-to-test mapping is appropriate and complete

It is not the user who is responsible for validating the suitability or appropriateness of the privacy enhancing infrastructure of the service provider, but a trusted third party. The user will only be responsible for checking that third party seals are current and valid and accessing the trustworthiness of the vouching party.

Trust has been moved from the certificate provision service (which converts certificates to the required format and adds in privacy-related meta-level information) to appropriate third parties that validate the deployment of privacy enhancing mechanisms.

In this way the end user can establish trust based on the reputation of the TTP, while the service provider can benefit from this trust relationship that has already been established, or has a better chance of growing stronger due to the fact that TTPs are trusted more than businesses, since trust is a TTP's business and so it is taken very seriously.
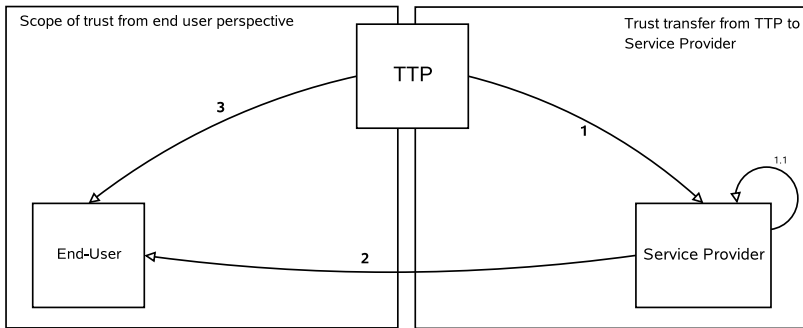
Fig. 16.7 The trust chain

To illustrate how trust is introduced into our solution we refer to Figure 16.7.

The arrow labelled 1 encapsulates the verifications, outlined in the bulletlist above, performed by the TTP to ensure trust in the service provider's back-end and how this translates to their privacy policies.

Once this has been done the TTP transfers a trust token to the service provider to display along with their privacy policies as well as with their policy validation results. The service provider will ensure that they display this trust token whenever it is applicable. This is depicted by arrow 1.1.

Once the end user has asked for privacy policies and\or verification results the trust token is transmitted to the end user. This stage is labelled as 2 in Figure 16.7.

Finally, the end user must now verify that the trust token is valid and intended for this set of results and the privacy policy under scrutiny. The end user can do this via a privacy seal verification scheme, such as one described in [MIT04]. Once the end user has checked the validity of the trust token they can then be assured that the results, whether positive or negative, are correct and worthy of trust.

It should be noted that a service provider does not need to have their entire back-end scrutinized by a TTP, only those parts that are relevant to PII storage, processing, or access and that are to be incorporated into their privacy policies. This way a service provider can roll out an incremental privacy enhancing program in their enterprise without having to roll it out all at once.

Privacy enhancing technologies (PET) such as those employed in PRIME, and related systems like trusted computing and infrastructure introduce best practice into the business's processes and infrastructure. If deployed and utilized correctly their presence can greatly boost the privacy capabilities of the business and ease the enforcement of clauses present in the privacy policy as well as allow for more. In the minds of consumers, a TTP verified PET shows that the business is serious about privacy and helps to build trust.

Also worth noting is the fact that the TTP do not have exclusivity and that both the service provider and end user can utilize any number of TTPs. From the service providers perspective they can use a hierarchy of TTPs to validate parts of their infrastructure, if that makes sense. For the end user, they can choose who to use by selecting a list of TTPs that they trust. This allows the end user to only trust privacy policies that are verified by a TTP that they trust and not one chosen by the service provider, thus empowering the end user more.

As an interesting foil to the discussion so far on trust, it is interesting to note that sometimes too much assurance information, especially when it is not well understood, can cause consumers to have a lower level of trust than if no information was provided. This should diminish as users' understanding of the technology increases and with it their appreciation of trust as a function of the assurance information being provided.

16.2.8   Extension to B2B Scenarios

Although the main purpose of PRIME is to empower individuals in protecting their privacy in customer to business (C2B) scenarios, our system is not limited to this type of usage, and indeed it is in individuals interests if a similar approach is used to protect their data if it needs to be shared amongst service providers. The techniques above can be used from user to SP, but also SP-SP to ensure data that is shared with third parties is treated accordingly.

Thus, businesses to business (B2B) and government to business use cases are also possible. As long as the proper protocols are in place our component can provide assurance information to any entity about any other entity. In addition to the PRIME integrated prototype (IP), we built a simplified prototype using the assurance control component to illustrate its usage within a B2B scenario.

In our scenario a business may wish to ensure that any customer PII they release to their partners will be handled accordingly. Assurance control may be part of a business's compliance process and ensuring that all PII sharing was responsibly carried out would mitigate the risks associated with PII leaks and misuse. Both PRIME IP and this demonstrator utilize the same implementation code, only differing in the way that it is deployed and the graphical user interface, which is external to the Assurance Control component.

We have also implemented a feedback form which allows users and administrators to give feedback to a checked party. For example, if the absence of

evidence for important properties caused them to abandon the transaction a
user can say this on the feedback page.

16.2.8.1   Resellers

Often, in practice, what seems like user to servide side interaction is actually
user to reseller to third party interaction. For example, if you buy an item
from Amazon, it might be that the item is actually being sold from the third
party shop 'SellIt'.

In this type of situation, it is not enough for the client to just check the
policies of the reseller — somehow the behaviour of the third party needs to
be checked also. Otherwise, Amazon might get certified by the BSI that the
assurance control policy 'always protects stored data using encyryption' holds.
However, it might be that SellIt is not interested in protecting data properly,
and doesn't use encryption. However, if the customer buys from SellIt via
Amazon, s/he will receive the assurance control policy from Amazon, but as
stated in the claim, the data will be transferred to SellIt. We would want to
avoid this type of situation.

The way in which this is done depends upon how the chain of trust works:
for example, the client could specify that the reseller must check that his
policies are respected on the third party, and trust them to do that (as with
the example considered in the previous subsection), or else the client might
want to know up front the policies of the third party.

For the process described in the previous subsection, the policies are as-
sociated with the specific service with which the client is dealing. In our case
here, the server may state (in its claim's DHP) that the data will be trans-
ferred to some other server. The client then may wish to check up this server
as well. Technically, this can be done using the same process as is used in
the simple case, but requiring either an additional assurance control loop of
checking to be made between the user and third party, or else between the
reseller and third party.

16.3   Comparison with Related Work

Our research is novel in several aspects in comparison with prior work in
this area. Steps towards the provision of more assurance to people on privacy
have been made by various privacy seals providers and verifiers [CC00]. This
approach provides users with general purpose information about the confor-
mance of a service provider or an enterprise with certified, privacy compliant
processes when handling and managing PII data. However these approaches
do not take into account specific, fine-grained requirements, needs and con-
straints dictated by individuals.

The usage of recommendation mechanisms [Res97, Net04] — based on peo-
ple sharing evaluations of enterprises' behaviour — is another well-explored

approach for dealing with trust matters. These mechanisms can also be used to evaluate enterprises' compliance to privacy and, as a side effect, have an impact on the perception of the trustworthiness of an organization. This approach is complementary to the problems the author wants to address. Related complementary work includes [Net04], which describes how a trust index of a CA may be computed.

As already discussed in Chapter 13 above, in our work we employ privacy practices that can be deduced from the W3C EPAL [IBM04] and P3P [Wor02] specifications and that implement the philosophy of recent privacy legislation. P3P specifications allow people to describe their privacy expectations and match them against the level of privacy supported by an enterprise. This helps shape people's trust in enterprises. However P3P only checks if their expectations are matched against promises made by the enterprise, and does not provide mechanisms to check and prove upfront compliance with fine-grained constraints.

Chapter 13 has already considered the relationship between our policy representation and prior work. As mentioned in that section, the Platform for Privacy Preferences (P3P)[Wor02] is relevant background for our work. There have been many critiques of P3P such as [Clab, Ack04, HJW02, Claa]. We shall ignore politico-economic arguments and focus on how our solution differs from P3P, the gaps it fills in, and how P3P could be used within the system we have implemented albeit with changes to its role. Expressing privacy concerns in P3P is done by defining statements in a machine readable format. Although there are editors[Claa] that help with this process, there are two problems that are not yet addressed.

1. The P3P language and editor are tools but the end user must know what they wish to express in the first place. They must know what their privacy vulnerabilities are and how to check if a website will mitigate those risks. Most users are naïve and would not be competent enough to express privacy concerns beyond vague statements.
2. Even with the prerequisite privacy knowledge, the definition of privacy polices must be in a language geared towards the facilitation of accessing PII based on conditions. Although useful it cannot capture other aspects of privacy adequately without losing some of the essence of what the end user intended. Our solution addresses these concerns by introducing standardised privacy clauses that are written in human readable form and are unambiguous, concise, and capture privacy concerns based on expert knowledge.

To ease the creation of polices templates are provided. The end user does not need to learn a language or an editor that requires knowledge of predicate logic.

As is the case with privacy seals, P3P cannot link the privacy practices expressed by the website and anything tangible on the back-end. This gap is where our solution introduces mechanisms to check that policies and the

technical realities of the website's infrastructure are coherent. Claims made in the privacy policies are backed up by capability checks as described in Section 16.2.5 and help to provide assurances that are missing from the P3P model.

Although P3P has its limitations, its strength as a robust policy definition language and logic model allows it to perfectly translate privacy clauses into machine readable form. The resultant privacy policy would have to be vetted by TTPs and also certified, or an intermediate layer could be introduced that would drive the policy editor to receive clauses and output machine readable policies. Since the clauses are defined and standardised the resultant XML would also be identical. In our model unique global identifiers are used to identify particular clauses, the drawback being that a unique identifier needs a lookup table to be maintained, whereas an XML policy would capture all the necessary information within itself. There have to be extensions to the present P3P vocabulary so that all aspects of privacy can be expressed.

Although there are obvious usability problems with P3P there are some efforts to make it easier to utilise. Projects like Privacy Bird[AT&] from AT&T and Privacy Fox[Ars] try to bring a simplified and more useful solution to end users. These projects provide a graphical face to P3P's policies and policy matching engine. End users can gage the how well their privacy preferences are matched by the website's privacy policy by displaying the results graphically. In the case of Privacy Bird an icon of a bird expresses how well the user and website polices agree. There are three levels, green for total agreement, yellow for partial, and red for complete disagreement. Our solution also utilizes graphics to depict the level of agreement between entities. The difference is that instead of just a single aggregate representation embodied by the bird icon we opted to give a more granular output so that the end user could have more context as to exactly what went wrong. The bird icon can show the mismatches between policies as well, if clicked, with the verbose human readable policy being displayed.

Privacy Fox[Ars] is a P3P extension for the Firefox browser; Privacy Bird[AT&] is Internet Explorer specific, and it adds the feature that it can summarise P3P policies for display to users in a table form. Although the resultant summarises are not as concise as clauses, they do have the advantage of being generated from the actual XML content of the privacy policy. Because the human readable form of the policies are written independent of the XML forms leading to discrepancies between the two this approach avoids this drawback. In our solution we have tried to take the best of both worlds and backed by usability tests incorporated both into a single HCI. Since clauses should be created as clear and concise statements in human readable form and results are shown as icons we have incorporated both Privacy Bird's and Fox's salient features in our Graphical User Interface (GUI). For further details about the user interfaces developed, see the later chapter on HCI.

16.4   Next Steps and Future R&D Work

Since clauses are the central privacy vector they need to be developed further from the select set that are being implemented now. They need to be more complex and recognise complex privacy needs of sophisticated users as well as laws and regulations that businesses must adhere to. They also need to be stated in such a way that is unambiguous in any language. Only the true essence of the privacy objective of the clause must be present in its description. This will be an interesting area which will require participation from law, business, and security experts to establish a coherent, effective, and simple language to define privacy issues and concerns.

At the moment the service provider depends on in-house security expertise or third party advice to implement and deploy privacy mechanisms. This dependence on security expertise could be avoided if the clauses themselves provided a set of tests that a service provider had to conduct. It could cut out the third party completely and move the reliance on to the PRIME system itself rather than third parties. The obstacles to resolving this are that service side topologies are not well understood and providing a generic yet robust enough set of tests that would be applicable everywhere is a difficult thing to do at present.

At present the client and service provider have no way to negotiate a privacy policy. The development of this functionality will allow fluid and mutable privacy policies being created on a per transaction basis. The value in this is that we believe the client would feel more empowered if he/she had some hand in creating the policy that they will accept.

Currently tests are implemented by the service provider. This takes time and effort and should be done correctly the first time round. For this reason third party security experts are required to ensure that is the case. If there is a change in the service provider's infrastructure these tests have to be recalibrated and reconfigured. This makes a dynamic set up hard to accomplish. To alleviate such constraints, agents that are connected to a central authority, internal or external to the business, can reconfigure themselves and adapt to changes in the topology and other logical influences. This can be either automated or controlled manually. This would reduce the effort of setting up and maintaining the assurance tests as well as give service providers a better overview of their systems and the ability to reflect changes in their privacy policies instantly. See [PA09] for an implementation of this approach.

16.5   Conclusions

Currently, end users do not have much control over how their PII is utilized by businesses, government, or healthcare. The most they can assume is that the organization will adhere to sound privacy principles. These assumptions are based on the belief that organizations are concerned by media publicity

and privacy regulations and laws and will take steps to act responsibly. These assumptions can be off base and the wary person would like to have more concrete information, or assurance, that their PII is going to be treated in a responsible manner that will not harm their present or future. Our solution aims to provide this assurance information, in a manner that gives the end user more control over the process, more trust in the system, and simplicity of use.

We have shown how a common standardized privacy clause pool would help communicate end user concerns as well as service provider promises. With the clauses forming policies we have designed a mapping framework that would allow high level clauses to be mapped to back-end technology that would abstract the complexity away for the end user and at the same time allow the service provider flexibility in how they implement and manage their infrastructure. Finally we have shown how trust is injected into this system through trusted third parties and their role in establishing a trust chain. This allows end users to form their own trust relationships with TTPs independent of service providers depending on their preferences and experiences.

In summary, this chapter describes the mechanisms used within the PRIME system for providing assurance information and building trust in privacy practices of businesses and other entities whilst being practical for deployment in current infrastructures.

17

Security/Trustworthiness Assessment of Platforms

Stephen Crane and Siani Pearson

HP Labs

17.1 Introduction

Platform assurance is a special case of establishing trust. In this chapter we give a generic assessment of trust, followed by an assessment of the impact of computer systems in relation to online trust. After this, we discuss trusted technologies that are currently deployed, and exactly how these may enhance trust and privacy. Finally, we explain how the PRIME Platform Trust Manager operates and can enhance trust.

Some related concepts have already been introduced in previous chapters, notably Chapter 16, where trust is discussed in relation to system policy compliance checking, and Chapter 9, where the trust negotiation concept is mentioned.

17.2 Assessment of Trust

In this chapter we discuss trust, and in particular organisational trust. Trust is both subjective and objective, and based on tangible organisational assets, i.e. platforms and the assurance that they offer, and on organisational practices, i.e. openness and guarantee. Trust is normally based on pre-agreed or negotiated understanding. In fact, agreement, possibly through negotiation, is fundamental to the process of establishing trust.

17.2.1   Trust in an Organisation

There exists a spectrum of possible identity management options (see Fig. 17.1). At one extreme there is the situation where the user adopts the approach of not releasing any personal identifying information at all. Instead, the user provides the recipient with information that has passed through some form of anonymiser. This is the approach to privacy that DRIM and Idemix can easily support.

Ideal position   Current position

True Anonymity ⬅⬛➡ Unrestricted Release

Fig. 17.1 Privacy spectrum

At the other end of the spectrum is unrestricted release of identifying information. This approach potentially exposes personal information to the greatest level of abuse, but is common practice nowadays for most commerce and services-based interactions.

The anonymising approach could be considered the ideal, and one to which PRIME should aspire. However, reflecting on the PRIME Project Description, one of the key goals of PRIME is to develop solutions that satisfy market-driven real world viability. Whether the world of commerce is able and willing to adapt existing practices and procedure to the extent that some anonymising techniques demand is still unclear. Furthermore, it is doubtful that a completely anonymous approach can be taken for many scenarios, e.g. healthcare and travel, where personal information simply must be divulged. Here techniques are required that protect personal information when linkage between some of the information and the owner cannot be prevented.

These possibilities present PRIME with opportunities to explore the challenge of true anonymity and examine other options at points along the spectrum, moving away from unrestricted release.

One final point worth noting is that during a typical interaction several different approaches may be required. For example, a user may begin anonymously and progress through to partial or full release of identifying information depending on how the interaction develops. An example is where a user requests advice about general medical care and (presumably happy with the advice) asks for more specific information based on personal symptoms.

17.2.2   Trust

Users want to be able to release personal information in the confident belief that it will only be used in the way the user intended. Providing this assurance is the key to demonstrating trustworthiness. For most situations, the trust that users place in an organisation is a mixture of technological trust and social trust. In many situations it is possible to manage technical trust by minimising risks using threat/vulnerability models. Social trust—the trust we place in another human—on the other hand, is very much more difficult to understand, measure and control.

Except for a handful of niche applications, technology and humans interact to affect outcome. On the whole, trust is limited to a belief that (say) an organisation will fulfil a request. There is usually limited evidence to support this belief other than possibly a contract that is only enforceable in specific circumstances. One way to understand trust better is to consider the nature of the participants. On the one hand there is the deceitful recipient who, if sufficiently motivated, will be able to circumvent controls (not always technical). This is a difficult category to deal with unless we can separate system and human trust.

Another category is the recipient who sets a high standard of business conduct and wishes to demonstrate this in order to provide differentiation from other less scrupulous recipients. This is an interesting category for two reasons: 1) the division between system and social trust is of less concern to the user; 2) this type of recipient probably represents the attitude of most major organisations. The latter are organisations that have valued brand and reputation, and are keen to show users that they can be trusted even if they cannot present indisputable facts that support their claim.

Of course, even the best-intended organisations make unintentional mistakes. These organisations would most likely welcome solutions to help them keep in check and reaffirm their own trust in their systems.

17.2.3   Determining Trustworthiness

Assuming the situation where an organisation is basically trustworthy but wishes to provide further evidence to this effect, the user can measure trustworthiness in the following ways as outlined next.

17.2.3.1   Trustworthiness of Services-Side System

Knowing that an organisation has adopted state-of-the-art trust technologies can be an initial sign to the user that the organisation intends to be true to their word. Today, state-of-the-art trust technologies mean a TPM (Trusted Processing Module) that provides:

- A reliable third party endorsed stable identity
- Originator non-repudiation achieved through TPM-controlled signatures

These requirements can be achieved by equipping a server with a TPM, endorsed by a Trusted Third Party, and building the functionality to allow 1) remote interrogation of the TPM by the user, and 2) automatic signing of acknowledgements and other information intended to convince the user that their wishes are being fulfilled.

In practice, the systems that support services offered by an organisation will be much more complex than a simple peer-to-peer arrangement. Whilst these systems may be built on TPM and future trusted platform technologies, techniques for forming an aggregated measure of trust across multiple heterogeneous systems that process personal information still need to be researched.

17.2.3.2   Trustworthiness of the Organisation

Trust in an organisation is built up over time, based in part on past interactions. Evidence that an organisation is willing to commit to an intended action, possibly in the knowledge that to not do so will incur penalties, is a useful sign of good intentions.

Typically, the user would either review or present the terms under which the interaction will take place (i.e. a policy or contract). Once accepted, these terms are binding to some degree. As required, the user reviews the interaction and compares outcome against the contract, particularly where the terms specify several points in the process where an assessment can be made (c.f. project milestones).

This leads us to a process with clearly definable steps:

- Policy/contract comparison between user and organisation
- Fulfilment (by the organisation)
- Checking (by the user)
- Opinion forming (by the user — essentially retention of evidence to aid trust evaluation during future interactions.)

The proposed approach differs from existing approaches (e.g. P3P) by providing feedback to the user and indeed involves the user / user's system in the process of active comparison and management The process can be presented diagrammatically as shown below.

The overall similarity between the diagram of Fig. 17.2 and the architecture of PRIME should be clear, except for the requirement on the user side of functionality to compare policy, and check and record status. In addition, user-side functionality that provides policy generation, proactive checking (that is, not relying simply on the organisation notifying the user of the status of an interaction), and presentation to the user of an aggregated and meaningful trust assessment is required.
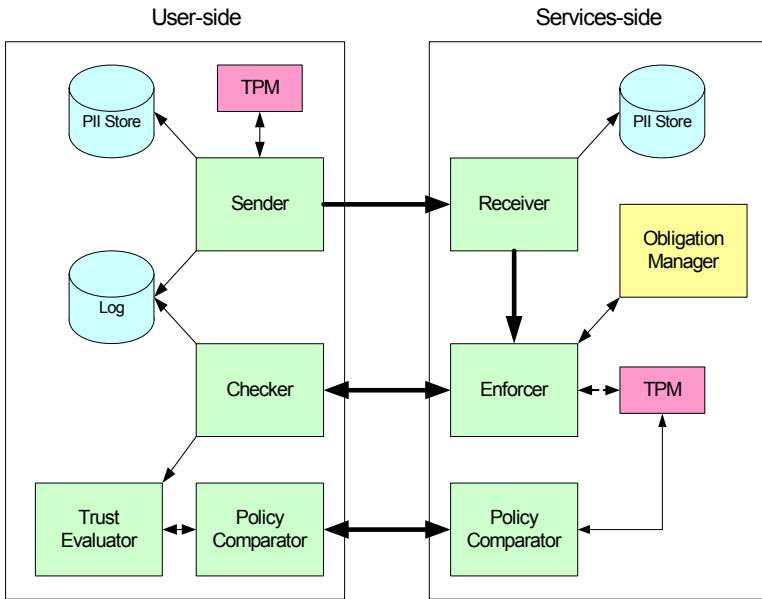
Fig. 17.2 Organisational trust

17.2.3.3   User-Side Trustworthiness

Whilst the user is concerned about the trustworthiness of the services provider, the user must also be able to trust their own system to hold their personal information securely. Assuming that the user is the only person with legitimate access to the system, trust is based solely on the technical merits of the system. Again, taking the TPM as the state-of-the-art technical security solution, the functionality to be supported by the TPM should include:

- Granting a user authorised access to personal information, i.e. identification and authentication of the user.
- Secure storage of personal information and/or the cryptographic key(s) used to control access to personal information.
- Generation of random seeds.
- Additionally, the TPM permits the generation/presentation of pseudonymous identities that may support or supplement credential management schemes like DRIM and Idemix.

Many users are likely to find the task of managing trust too difficult because it requires specialist skill and knowledge. Ways of providing help and support to the user through UIs, warning mechanism, best practice advice, etc. will need to be deployed to help users check/preserve their platform's trustworthiness and avoid making decisions that could compromise their platform.

These are ambitious goals, involving long-term research, but we can start by leveraging the functionalities provided by TPMs and trusted platforms.

Looking further into the future, and the evolution of ambient services and devices, managing trust on the user side goes beyond the relatively straight-forward gatekeeper role that we see here to that of an agent. Imagine the situation where a user has a need for particular service, and instructs their personal system to look for the most appropriate services on offer. Part of this process could involve the automatic release of personal information about the user. How can the user be confident that their personal system is acting in the best way to preserve their privacy?

17.2.4   Summary

By concentrating on the specific situation described, i.e. where the organisation is essentially trustworthy but needs to be able to demonstrate this publicly, we can provide users with the means to differentiate likely trustworthy from untrustworthy parties to which the user intends to release personal information.

State-of-the-art TPM technology (e.g. developed by the Trusted Computing Group – TCG) provides the foundation on which to create protocols that provide evidence of intent to comply with user wishes, a means to resolve discrepancies and the ability for the user to form an opinion about the trustworthiness of a recipient in those situations where the user is obliged to release personal information.

Whilst TPM and trusted platform technologies provide a foundation for trust services, further research is required to develop a better understand of aggregated trust measurement and the complexity of communicating trustworthiness to a non-specialist user.

Overall, the approaches to trust management described in this paper provide the user with increased confidence when releasing information to other parties, and improves on the current situation of unrestricted release by offering options along the identity management spectrum.

17.3   Assessing the Impact of Computer Systems in Relation to On-Line Trust

17.3.1   Analysis of Online Trust

As we have considered in previous chapters, trust is a complex notion for which there is no universally accepted scholarly definition, and which includes temporal, risk, delegation and dynamic aspects. Additional issues that relate to on-line trust include:

- *Brand image.* Reputation is perhaps a company's most valuable asset [Nis99] (although a company's reputation may not be justified), partly because trust is a better strategy than power games [Kum96]. Brand image is associated with trust and suffers if there is a breach of trust or privacy.
- *Delegation and provision of assurance information.* Due to a lack of information and time, together with the huge complexity of IT security, it is impossible for users of IT products to identify the level of security offered by individual products. They need to rely upon the reliability of a product being assessed by experts via evaluation and certification procedures, such as using criteria catalogues (e.g. the 'orange book', ITSEC, Common Criteria in ISO/IEC). The idea is that if a certain assurance level is reached after testing and evaluation of a product, it is worthy of trust being invested in it. Such delegation of trust underpins security certification and privacy seals.
- *Security and privacy.* Enhancing security will not necessarily increase trust, but it is an important enabler and can do so. Some would argue that security is not even a component of trust. For example, Nissenbaum argues that the level of security does not affect trust [Nis99]. She argues that security is increased in order to reduce risk, and not to increase trustworthiness. However, we would argue that, according to the situation, security may increase the level of trust, decrease the level of trust or indeed be neutral as Nissenbaum suggests. An example of increasing security to increase trust comes from people being more willing to engage in e-commerce if they are assured that their credit card numbers and personal data are cryptographically protected [Gif00].

  Note that there can be a conflict between security and privacy. For example, some methods of enhanced authentication can result in privacy concerns (such as manufacturers' issue of identification numbers associated with networked devices). Indeed, in order for users who value privacy highly to regard a computing system as trusted, it is important that increased security does not have an adverse effect on privacy.

  For further general discussion related to trust in Information Technology, see [ACM00, CT01] and various recent research studies that analyse trust in relation to the e-commerce domain [Che99, Egg98, FKH00].

17.3.2 How On-Line Trust Is Underpinned by Social and Technological Mechanisms

When assessing how trust may be increased by computer systems, we see it as helpful in distinguishing between persistent and dynamic trust, and between social and technological means of achieving such trust:

Persistent trust is trust in long-term underlying properties or infrastructure; this arises through relatively static social and technological mechanisms. Social mechanisms, behaviour and values contributing to this include sanctions, assurance and vouching (including seals of approval): such examples

are of infrastructural mechanisms that may vary over time, but in general are relatively stable. Technological mechanisms include underlying security infrastructure, well-known practices and the technological features corresponding to static social mechanisms; these can involve the following, for example:

- Certified hardware (for example, tamper-resistant hardware)
- Protocols
- Certified cryptographic techniques
- Assurance
- Other security features
- Audit and enforcement

Dynamic trust is trust specific to certain states, contexts, or short-term or variable information; this can arise through context-based social and technological mechanisms. The content of social mechanisms would be liable to substantial change at short notice, such as brand image, look and feel, reputation and history of interactions. The technological mechanisms give confidence that a particular environment or system state is trusted (at a given time, for a particular purpose). A system's behaviour can change according to a given context, and in particular if it has been hacked, and in some cases system behaviour can be driven by policies (dictated by people, business needs or even malicious people) that change over time. For example, dynamic trust could be affected by the following information being divulged:

- A particular system has been compromised (for example, spyware is running on it)
- The location of the system or user has changed
- Software is in a certain state
- Policy enforcement has not been carried out

The relationship between these categories can be complex, and in particular the distinction between persistent and dynamic trust should be viewed as a continuum because there is not always a clear-cut distinction between these categories. For example, recommendation could be considered to be inbetween these categories as it is in general fairly static but could still change in the short term.

The focus of this chapter is on a subset of persistent (social and technological-based) and of dynamic (technological-based) trust. Both social and technological aspects of trust are necessary when designing online systems, quite apart from additional social guarantees of privacy and security.

17.3.3   Summary

Trust is a complex notion and a multi-level analysis is important in order to try to understand it. There are many different ways in which on-line trust can be established: security may be one of these (although security, on its own, does not necessarily imply trust [Ost01]). When assessing trust in relation

to computer systems, we have distinguished between social and technological means of providing persistent and dynamic trust. All of these aspects of trust can be necessary. Persistent social-based trust in a hardware or software component or system is an expression of confidence in technological-based trust, because it is assurance about implementation and operation of that component or system. In particular, there are links between social-based trust and technological-based trust through the vouching mechanism, because it is important to know who is vouching for something as well as what they are vouching; hence social-based trust should always be considered.

Mechanisms to provide dynamic technological-based trust need to be used in combination with social and technological mechanisms for providing persistent trust: as we shall see in the following section, if software processes provide information about the behaviour of a platform, that information can only be trusted if entities that are trusted vouch both for the method of providing the information and for the expected value of the information.

17.4 Deploying Trusted Technologies

17.4.1 Trusted Computing Technology

Trusted computing solutions, like those being developed by the Trusted Computing Group (TCG) [BCP$^+$02, Tru03], can address the lower-level protection of data. The TCG is an organization set up to design and develop specifications for computing platforms that create a foundation of trust for software processes, based on a Trusted Platform Module (TPM) [Tru06, Tru03]. This is a cost-effective tamper-resistant cryptographic hardware component within a platform that acts as a root of trust.

TCG technology can be used in order to address the threat of fraudulent access of locally stored data, by enabling strong encryption and strong protection of keys. Low cost TPMs are becoming commodities in business computing devices (PCs, laptops and other systems) further accelerating TCG technology adoption.

For the time being trusted platforms (TPs) are used mainly to protect keys and other platform secrets via the TPM and to execute secure cryptography operations. Allied protected computing environments under development by certain manufacturers and open source operating systems such as Linux can support TCG facilities further: Intel's Vanderpool Technology (VT), Trusted eXecution Technology (TXT) hardware and chipset modifications; Microsoft's leverage of TPMs within their Vista and Longhorn Server OSs. These different trusted computing implementations are all TPs since they accord to the same underlying philosophy and basic principles of operation, as espoused in [Tru04]. In the longer term, as specified by TCG, trusted computing will provide cryptographic functionality, hardware-based protected storage of

secrets, platform attestation and mechanisms for secure boot and integrity checking [Tru03].

Trusted computing addresses some central concerns of people using PCs: it protects data that is stored on those machines (even while they are interacting with other machines over the Internet) and it aims to put everyone in the position where they can feel confident that they can:

- Protect their data
- Find out whether their platform is in a trustworthy state (i.e. its integrity has not been compromised)
- Have the means to decide whether it is reasonable for them to trust other platforms

17.4.2   How Trusted Platforms Can Provide Persistent and Dynamic Trust

Broadly speaking, the view taken by the proponents of trusted computing (see for example [Tru04]) is that we can think of something as being trusted if it operates in the expected manner for a particular purpose, or can be relied upon to signal clearly if it does not. The TCG definition of trust is that something is trusted "if it always behaves in the expected manner for the intended purpose" [Tru04]. A similar approach is also adopted in the third part of ISO/IEC 15408 standard [Int99]: "a trusted component, operation or process is one whose behavior is predictable under almost any operating condition and which is highly resistant to subversion by application software, viruses and a given level of physical interference".

We believe that categorizing trust in terms of the analysis presented above helps in understanding further how TPs enhance trust.

- **Dynamic v. persistent trust.** Within a TP, a trust hierarchy operates such that such behavioural trust in the platform is underpinned by trust that the platform is at that time properly reporting and protecting information (dynamic trust), again underpinned by another layer of trust that that platform is capable of properly reporting and protecting information (persistent trust). Both dynamic and persistent trust are involved in the decision by an enquirer (either local user or remote entity) whether a platform is trusted for the purpose intended by that enquirer: if the enquirer trusts the judgment of the third parties that vouch for the system components, and if the platform proves its identity and the measurements match the expected measurements, then the enquirer will trust that the platform will behave in a trustworthy and predictable manner. The platform reports information to the enquirer to enable that decision to be made [Tru06], and analysing this requires intelligent application of cryptographic techniques; optionally, use could be made of a third party service to perform or help with this analysis. In reality, the enquirer might want to be reassured about a set of platforms running services, managing

their personal data. Analysing such composite assurance is an issue we have addressed within Prime.

In relation to the aspects of trust that were discussed above:

- **Temporal aspect.** Some trust is based on our use of similar technological products beforehand, and our history of interactions with companies (see the discussion about brand image below). In the shorter term, trust sustainability should be addressed: in order to maintain a trust relationship between service requester and provider over time, or at least until a service is completed, the service requester may periodically re-challenge the provider to check the latest integrity metrics. The analysis required may need to take account of other factors too, such as time and history. Another approach is to have enhanced trusted software on the provider platform that monitors any changes to the platform state against pre-registered conditions provided by the service requester and notifies the requester if the changes impact the conditions [YC04]. This can be more efficient but requires additional initial setup and infrastructure; moreover, it can potentially lessen trust and security if the root of trust for reporting is no longer the TPM hardware chip.
- **Risk aspect.** Risk can operate at various levels, including business with strangers being risky (no less so for business partners online than it is off-line [Jup01]) and threats from hackers. Trust in a TP ultimately reduces to trust in social entities, which involves risk. On the technological side, there are risks arising from the necessity to reduce complexity of platform state during analysis for practical reasons (and hence a focus on checking only selected integrity metrics) and privacy or security risks that trusted computing does not protect against – for example, unauthorised keystroke logging. The first generation of TPs only provides a protected storage capability – it does not expose the full functionality described in the TCG specifications and can only be trusted to protect secrets in a certain way since there is no trusted boot process. TPs that do provide the full functionality described in the TCG specifications provide roots of trust for systems, but even so they do not provide a complete trust solution: instead, additional trust functionality should be built on top of them. In the short term, this must include security enhancements at the operating system level (as mentioned above), right up to trust management techniques (see for example [BIK03, GS03]). Even if all this functionality were provided in a system, no system is ever completely secure, so there is some risk, however unlikely.
- **Delegation.** Trust in a computer system is underpinned by trust in individuals, in companies, and in brand names who vouch for the system. Lack of trust by some people in some entities involved in the production of trusted computing is a reason for them to distrust the technology as a whole (see for example [Yun03]). The (persistent) social basis for trust is that trusted third parties vouch (a) for the mechanisms that collect and

provide evidence of dynamic trust as well as (b) that particular values of integrity metrics represent a platform that is behaving as it should. In essence, delegation is centrally involved: certain third parties are prepared to endorse a platform because they have assessed the platform and others are willing to state that if measurements of the integrity of that platform are of a certain value, it can be trusted for particular purposes.

In order to do the former, an endorsement key is embedded into the TPM. The public endorsement key is signed by the manufacturer and published in the form of a digital certificate. Social trust is used to recognise a specific genuine TPM: you trust a specific TPM because it is an assertion made by the trusted manufacturer that produced it. In a similar way, other elements of a TP also have certificates, and delegation of trust to authorities is needed in order to provide such certification material: TCG provides this in conformance with the Common Criteria.

- **More on dynamic aspect.** In order to know whether a platform can be trusted at a given time, there are processes in a TP that dynamically collect and provide evidence of platform behaviour. These processes carry out measurement and provide a means for the measurement method to show itself to be trustworthy. When any platform starts, a core root of trust for measurement (inside the BIOS or the BIOS Boot Block in PCs) starts a series of measurements involving the processor, OS loader, and other platform components. The TPM acts as a root of trust for reporting and dynamically stores and protects against alteration of the results of this measurement process, as well as reliably cryptographically reporting the current measured values. To find out if a service executes properly (in a dynamic way) you check these measurements against values that have been created and signed by someone that you trust, as discussed above.
- **Brand image.** Brand image can be leveraged to sell trusted systems. Someone's willingness to carry out business with a TP will depend on the intended use and on the level of trust in the platform and the owner of the platform. In particular, the manufacturer of a platform is visible to a third party communicating with that platform. For example, Original Equipment Manufacturers (OEMs) can exploit their reputation for quality to make their platforms the preferred solution for business-critical services.
- **Security and privacy.** Trusted computing is designed to provide enhanced security at an affordable price. Trusted computing is also designed to provide this security in a privacy-friendly manner – for example, it provides pseudonymous or anonymous attestation identities: see the following section for further discussion of this issue.

17.4.3   Summary

In conclusion, answers to questions about technology-mediated trust involve a combination of technology and also (changing) human attitudes and behaviour. In order to determine whether a system is trustworthy, we have to

ask whether we have assurance that the system will behave as it should and also whether we trust the people behind the technology. TPs help in doing this, but note that still we trust these people if we believe that they will not exploit their potential to hurt us. By the mechanisms described above, the next versions of TPs will aim to provide a root of trust for other trust service technologies.

17.5   Use of Trusted Computing to Enhance Privacy

17.5.1   Introduction

As we have seen in the previous section, the term 'trusted computing' can apply to a range of similar technologies for developing 'more secure' computing platforms in a cost-effective way. At best, this is only a partial solution to the privacy problem, because there are some business issues it is unlikely to address (such as unwanted marketing or being forced to give personal information to obtain a service). In addition, there are numerous classes of technical problems that are not addressed by trusted computer technology, such as spyware, adware, keystroke loggers, network sniffers, and wireless interception. Moreover, in being a technical approach it does not help address relative privacy threats, for example which may involve employer rights, local legal frameworks, and so on, or involve abuse of personal information that is provided voluntarily.

Trusted computer technology has been seen by some as a threat to privacy and freedom; however we argue that it also has tremendous potential for enhancing and protecting privacy. This is an important aspect of this new technology that has not been fully appreciated to date. For example, trusted (computing) platforms can be used to provide many pseudonymous identities, each of which nevertheless inherits the trustworthiness of the basic platform, hardware protection for secrets and independent mechanisms for verifying the trustworthiness of users' own systems and those they interact with. In this section we assess counterarguments that are based on misconceptions, and highlight where there are still further privacy-related issues to be resolved before such technology should be applied.

17.5.2   How Trusted Computing Platform Technology Can Enhance Privacy

Trusted computing provides the following key features, based on the basic functionalities described in the previous section, that provide building blocks for privacy:

- protection for users' secrets: 'protected storage' functionality binds secrets to a platform and can even prevent the revelation of secrets unless the software state is approved.

- potential for remote trust: users or enterprises can recognise that a platform has known properties and identify that a system will behave as expected. The technology can for example allow users to trust the platform in front of them to handle their banking or medical data.

In essence, these features allow privacy-protecting software, like any other software, to be more certain about the software environment under which it is running and to utilise more secure storage for private data and secrets.

Lurid media reports may refer to a TCG device as a spy-in-the-box, covertly gathering and reporting information on the user to the unprincipled giant corporation, and incidentally ensuring that only their products are compatible with the system. But with appropriate industry-agreed safeguards the reverse can be the case: the device can guarantee certain privacy and security aspects in the on-line marketplace and be broadly compatible with all products and systems. It is therefore vitally important that appropriate safeguards are agreed and introduced.

17.5.3 Privacy Enhancing Safeguards of Trusted Computing Technology

Various fundamental privacy-enhancing safeguards are integral to the TCG specifications, as follows.

17.5.3.1 Owner Control

Ultimate control over activation and the functionality of the TPM must be given to the platform owner, and users can deactivate it if desired. Also, the TCG organisation does not endorse any particular supplier or certifier of TCG-compliant hardware or software, nor does it provide such a role itself; notably, the choice of software to be run on a platform is entirely under the owner's control.

17.5.3.2 TCG Pseudonymous Identities

The TCG specification deliberately does not include attempts to identify which platform is making statements or communicating; instead it prefers to allow the use of attributes or credentials and gives the communicating entity better reason to trust these attributes. There is no need to have a single stable identity across transactions, thereby avoiding the privacy-related pitfalls of having a unique identity (which has affected some previous technologies). The only cryptographic key that is permanently associated with the TPM (the endorsement key) is designed to never sign or encrypt data, in order that an outside observer will not see anything traceable back to that TPM, and is only used to decrypt the response from a Privacy-CA within the TPM identity creation process.

TCG platform attestation therefore protects users against correlation and tracking of their activities as follows:

- TCG provides for the TPM to have control over "multiple pseudonymous attestation identities"
- There is no need for any identity to contain owner or user related information, since it is a platform identity to attest to platform properties
- The identity creation protocol allows the choice of different Privacy-CAs to certify each TPM identity, and thus helps prevent correlation of such identities

The origin of a specific identity cannot be tracked further, except by the Certification Authority (CA) that issues a certificate for that attestation identity. So appropriate selection of CAs enables the owner to control traceability from an attestation identity to the certificates that attest to a specific TPM and a specific platform. Identities can only be correlated with other identities by the CA that certifies these identities – and the owner chooses that CA (admittedly, negotiation with the communication partner may be needed to find a CA that is trusted by both parties). So the owner can choose a CA whose policy is not to correlate identities, or whose policy is to correlate identities, according to the wishes of the owner. Different identities are used for different purposes and in particular, separate identities would usually be given to different users of the trusted platform. The TCG specification [Tru03] includes the option of using a zero-knowledge protocol to obtain platform identities; this enables a CA to issue identity certificates without having to see platform credentials.

17.5.3.3 Data Protection

Each user's data can be kept private such that even the platform owner or administrator cannot access that data without the necessary access data. Furthermore, the revelation of secrets can be prevented unless the software is in an approved state. (However, as mentioned above, only a standard protection against direct physical attack is provided).

The TCG specifications have been designed in full support of data protection legislation. For instance, if you are handling personal data, you need to have reasonable assurances about the software you are using that handles these data, and trusted computing helps provide this. As a platform for handling personal data, a trusted platform allows you to comply more effectively with the requirement to handle Personally Identifying Information (PII), or other types of sensitive personal data, in a suitably secure manner. By effectively isolating certain classes of data from the rest of the system, you can significantly reduce the risk that such protected data will 'leak' to, or be actively 'stolen' by, other software. In addition, you want to minimise the personal data that is revealed, and as we have seen the TCG specification deliberately

minimises both the potential for identifying a platform across multiple uses, and the use of identifying data.

17.5.4   How Such Building Blocks Can Be Used

We have seen that trusted computing provides useful building blocks for privacy, but additional mechanisms such as identity management will need to be used in addition in order to provide a complete privacy solution.

17.5.4.1   Properties and Enhancements

Broadly speaking, trusted platforms provide the following properties, which are useful for a range of services including electronic business, corporate infrastructure security and e-government:

1. recognizing that a platform has known properties, both locally and remotely. This is useful in scenarios such as deciding whether to allow mobile platform access to a corporate network and providing remote access via a known public access point.
2. identifying that a system will behave as expected, both locally and remotely. Again, this property can be exploited in order to allow mobile access to a corporate network only with appropriate security measures (e.g. firewall and antivirus requirements) or to verify the integrity of a service provider platform.
3. enabling a user to have more confidence in the behaviour of the platform in front of them. In particular, users can have more trust in a platform to handle private data. A trusted platform helps provide assurances about software that handles personal data or personally identifying data in a suitably secure manner.

Specifically, the following enhancements provide greater confidence in protection of private data:

- Increasing users' confidence about their data residing on the server: It becomes possible not only to store personal data on the server more safely but also for the data owner to restrict the conditions under which that data can be used (more specifically, to specify appropriate software environments in which the data can be used or exposed). An example of where this would be useful would be in healthcare, where, say, a patient may need to send details of medication to a doctor in order to receive advice, but they would want to be assured that this information would be adequately protected within the on-line medical system.
- Confidence in an appropriate form of data being disclosed: The amount and type of data disclosed could be dependent upon the perceived trustworthiness of the receiving party. For example, data could be generalised, sent with policy conditions attached or not sent at all.

- Checking for appropriate treatment of data: Enforcing privacy policies via an extension of TCG software verification: it is possible to check that there is appropriate technology for enforcing policies on a remote platform before making the associated information available. This is particularly relevant in multi-party scenarios where there need to be privacy-related controls over how sensitive or personal information is passed around between the involved entities. This could be the case in federated identity management (cf. arranging a holiday via a travel agent), or in online e-commerce involving multiple parties (cf. buying a book that needs to be delivered to your address).
- Preventing disclosure of personal data or secrets in an adverse software environment: Integrity metrics relating to the server platform could be sent to the user when authorisation was needed, so that the user could assess trustworthiness of the server platform before making a decision of whether to authorise the key. TCG technology provides a special wrapping process that allows the authorised owner of a secret to state the software environment that must exist in the platform before the secret will be unwrapped by the TPM. This allows prevention of disclosure of customers' data and other confidential data or secrets stored on the server in an adverse software environment. Such an environment might masquerade as a safe software environment, and in particular enable undesirable access to, alteration of or copying of the data.

TCG technology not only allows existing applications to benefit from enhanced security but also can encourage the development of new applications or services that require higher security levels than are presently available. Applications and services that would benefit include electronic cash, email, hot-desking (allowing mobile users to share a pool of computers), platform management, single sign-on (enabling the user to authenticate himself or herself just once when using different applications during the same work session), virtual private networks, Web access, and digital content delivery. In general, applications which would benefit from trusted computing technology, as opposed to other secure hardware solutions with enhanced OS, are those that need the properties described above, provided in a cost-effective manner without the need for cryptographic acceleration or physical security.

In order to make use of trusted computing, it is not necessary that all the communicating computers are trusted computers. Firstly, it is not necessary to be a trusted computer in order to challenge a trusted computer and to analyse the resulting integrity metrics, although it helps to verify this on a trusted computer, since one might have more confidence in the result. Secondly, it is not necessary to use a trusted computer to produce pseudonymous identities (although the provision of such identities is a useful feature of trusted computers); in any case the trusted hardware can be used to generate new key pairs such that the private part is never exposed outside the trusted hardware and identity certificates can be created that bind the public part

of such key pairs to privileges, authority or attributes of users. Trusted computers provide additional useful functionality in the form of protected storage and remote verification (aka integrity checking). Hence, depending upon the service model or application, it could be of benefit to have the clients and/or severs as being trusted platforms.

This technology can be used in a wide variety of scenarios (see for example [BCP+02]). It can also be extended and integrated with other types of security and privacy-preserving technology. For example, imagine a home of the future in which every appliance is wirelessly connected. We may need to know not just how to identify the true device to which each component's output must be directed (e.g., not to a neighbour's house), but also the functionality of the receiving device (e.g. that it is not a video capture device that could be used to circumvent copy protection). Building systems in which we trust can provide these answers. Furthermore, if the information being shared between devices is highly personal (e.g. you want to show a friend your holiday photographs yet retain ownership of the photographs and control how they can be viewed), special privacy-preserving techniques need to be added on top of the underlying technologies that demonstrate trustworthiness.

TC can help provide such a solution. A user may decide to transmit personal or sensitive information from their trusted device only if the receiving platform can prove that it has an appropriate trusted environment in place. Similarly, an appliance with enough computing power could automatically make such a check about a trusted appliance. The receiving platform can prove it is a genuine trusted platform (using cryptographic attestation identities) and send integrity metric information to the requesting device, which will check whether the integrity metric values correspond to previously published values and decide whether the entities vouching for this information are regarded as trustworthy.

Privacy protection policies could be used to express the constraints to be checked, and can even "wrap" personal information after transfer. Platform-level enforcement of these policies can ensure that the user's data is treated in the manner that the user would expect, in the sense of being in accordance with their privacy policies.

See [SPC05] for additional examples of how trusted computing technology might be applied in the future, in such a way as to enhance users' privacy.

17.5.5   Potential Negative Privacy Implications of Trusted Computing

The type of techniques we describe can be applied to benefit users in a variety of situations. They can also be applied to benefit corporations, sometimes at the expense of users' freedoms. It is this latter case that is often highlighted (and sometimes exaggerated) within publicity about trusted computing, but there is bound to be a spectrum of opinion about trusted computing technology largely depending upon the standpoint or vested interests of the holders.

It is true that technical and legal solutions to such privacy issues are less well developed than for the enterprise space, largely because consumer issues are not the initial focus for TCG. However, for this very reason there is still time to engage in such a debate and it is important that we do so, particularly because privacy concerns are very relevant at the user level.

Much of the public discussion about trusted computing is demonstrably factually incorrect, or coloured by people's worst fears – hardly surprising when there is a new technology or initial lack of information. Popular misconceptions include that the technology cannot be disabled and that unapproved, uncertified and open-source software cannot run on a trusted platform.

This is not to deny that there are problematic aspects for the consumer space. Trusted computing does provide a building block for privacy, as considered above, but it is only a building block and not a complete solution. Most notably, when you apply for credentials, you might reasonably need to provide personal information (as in the case of insurance), and this could be passed on to others. Whether any personal data is required by a given Privacy-CA when it issues a pseudonym is a matter for that Privacy-CA to decide. The TCG specification makes no such requirement: national legislation may (in different jurisdictions) either prohibit or mandate such linkage.

Probably the most important objection to trusted computing that is commonly made is against customer lock-in via 'trusted' applications. As with any technology, and especially one which is powerful enough to provide high levels of protection against accidental and malicious disclosure, there is potential for commercial abuse in that a content provider or software supplier might require a user to provide excessive personally identifying data, or to use only 'approved' software that is not produced by competitors. The TCG specification does not give a controlling position to any particular commercial body, but this is only part of the solution. Any such abuse of market dominance needs to be prevented using relevant laws and agreements, including anti-monopoly and data protection legislation, and it is important that this does happen.

Concerns have also been voiced in the public arena about the potential use of trusted computing for:

- **remote censorship:** This should not be a major issue since TCG does not help at all with deleting selected files on other platforms and in order to find them, the owner of a trusted platform would have to agree to the appropriate measurements being made and in any case such a comparison could be easily subverted by storing the same content in a slightly modified way.
- **increased tracking of users, even to the extent of 'spying' on users:** Trusted computing does not make the situation worse, so long as 'trusted' source code were published to show that tracking mechanisms were not secretly hidden. This is one of several good reasons why TCG-compliant open source software should be encouraged and supported.

- **loss of user control through having keys on the platform that are controlled by third parties:** There is not a privacy problem in the endorsement key pair being generated within the TPM, as this is independently trustworthy and it is necessary to have some private key within the machine associated with an authorisation secret that can still be used in some restricted ways if the machine is compromised. The problem is rather that you need to trust that the key has been generated, or put in properly, and so we need to guard against the possibility of chip vendors storing the private key and modulus, even if they deny doing this.

There are two further issues related to privacy and personal choice:

1. If you have a platform that provides you with mechanisms to protect your information, should a third party be able to use these mechanisms in your computer to protect their information from you? This question is particularly contentious because of the issues that certain groups have with the use of Digital Rights Management (DRM), particularly in circumstances when the users affected do not want to be governed by such enforcement and may lose 'fair rights' usage as a result. For discussion of how fair use might be achieved when building DRM technologies on top of TCG, see [Eri02]. Note however that TCG does not address DRM directly and it is certainly possible to carry out DRM without TCG. Concerns by anti-DRM proponents would be greatly reduced if it could be guaranteed that in given situations output from applications belongs to the owner of the platform or user of the application rather than the application provider, and can always be accessed by the owner or user, using the viewer of their choice.
2. How will the open-source community self-certify software, and will owners of commercial data trust that software? The problem is essentially that if a data owner cares enough to protect his or her data, he or she would probably not want an arbitrarily modified software environment to operate on those data, yet should be able to choose to use open source products. Open source suppliers could certify certain software distributions, to say that these distributions will operate as they should, but this takes money and time, and restricts the revisions that can easily take place. Furthermore, there is the issue of whether owners of commercial data would trust such distributions to respect the use of those data. This issue is being addressed within the Open TC project [Con08].

17.5.6   Concluding Remarks

Trusted computing platforms are already available for purchase and several more types will be appearing over the next few years. These are designed to be a cheap, exportable and ubiquitous way of improving the security of personal, corporate and government data. However, trusted platforms are not yet intended for ordinary consumers. This section aims to highlight how trusted

platforms can, in the near future, provide building blocks for privacy and thereby help bring widespread benefits for consumers.

Trusted computing technology does not provide a complete privacy solution, even for those technological aspects of privacy that it can help enhance, but it could be used in combination with mechanisms such as identity management and privacy policy specification and enforcement (so that you can keep control over personal data once it has been released via the TPM, for instance). Hence there is a role in integrating this technology in some instances with PRIME technology. In addition, there needs to be appropriate interpretation of current data protection principles, for example regarding the treatment of personal information given when applying for TCG credentials.

Trusted computing technologies can be used to provide better data protection and consumer control over Personally Identifiable Information (PII) [Pea03]; however, similar technologies can be used for more controversial technologies such as Digital Rights Management (DRM). So, as with many other technologies, trusted computing could be used as a basis for applications or services that individuals might regard as desirable or otherwise. An analogy would be the telephone, which brings many benefits but can also be used for surveillance. Just as we would not think of surveillance as being the inspiration or a necessary part of telephony, we are liable to miss out on many potential benefits for individual users if we do not adopt the technology with appropriate safeguards. In this section, various examples have been given of how rich the potential application of such trusted computers could be and how this may benefit users in a variety of circumstances. As with many technologies, trusted computing could be used for constraining applications or protection of proprietary interests. As such uses are inconsistent with users' needs, we may expect both market and regulatory pressures to direct that its use will instead focus on enhanced privacy and the benefits that increased trust of computers can bring to a wide variety of users. In order to do this, technical mechanisms, standards and laws are insufficient enough by themselves. These need to be developed in parallel, and in addition a political framework is needed in which to guide operation of such solutions. This is work which is ongoing.

17.6 PRIME Platform Trust Manager (PTM)

Overall, the role of the PTM (or TM) is to provide evidence to support a claim that a process will be performed in the manner stated. The TM is present on both the User-side and the Services-side, and in both instances core modules of the TM component are present. A User calls on the TM when they need to know whether their local platform manages their personal information in an agreed manner or if the other party that they interact with (either the Services side or another User) is able and willing to uphold any data management request.

Measuring trust involves 1) assessing the capabilities of a system (typically the capabilities of the hardware and software components) and 2) providing an indication of past performance. For this to be an effective measurement, capabilities of the system must be linked to intended application, and past performance (which can include the way that another party operates their system as well as system qualities) is used as the basis of a reputation measurement (suggesting likely trustworthiness). The modules forming the TM component are shown in Figure 17.3. At the highest level the TM contains six modules: Trust Handler (TH), Trust Wrapper (TW), Platform Trust Status (PTS), Trust Real-time Monitor (TRM), Trust Communicator (TC) and Reputation Management (RM). The Trust Wrapper module provides the first level of abstraction from the hardware TPM (Trusted Platform Module). The TH handles queries from the IDM and PCC about trustworthy status and represents the first point of contact within the TM for any calling module. The TH calls on the TW, the RM and the PTS to gather information to form the response to these queries. On-going monitoring of the trust status of the platform is carried out by the TRM module, which raises alerts to the TH when a discrepancy arises. The TC module is responsible for aggregating trust indicators and presenting them to the User in an easily understood format.

Of these modules, all except for the TC and RM are considered core modules. The TC and RM modules are only appropriate on the User side and Services side respectively (but see also comment on trust aggregation in the section on the TC Module). An enquiry about the trustworthiness of a platform takes the general form "tell me what trust technologies you support". This enquiry is received by the TM from the application, having passed through the AI and the IDM. The TM responds to the User with a list of supported trust capabilities. With this knowledge the user can begin a negotiation at a lower level to agree which trust capabilities should be applied to specific pieces of personal information. For example, a Service Provider may support an Obligation Manager that in turn supports deletion of data. The User may want to apply this privacy technology to any credit card information that they provide to the Service Provider.

Since the TM will be requesting information from other components in order to formulate an overall opinion about trustworthiness, The Access Control (AC) component will be responsible for authorising access. The results of this enquiry enable a user to determine whether they should trust a Service Provider. Exactly how this information is communicated to the user, and how the computation takes place, is currently undecided but is likely to be by way of a simple graphical representation influenced by context (i.e. trust status will depend on the application). Computing trust will in part be the responsibility of the User (since it will involve the combining of social and technical indicators) by the process will be automated wherever possible, particularly where the User is unlikely to be able to comprehend the trust information they are presented with or where interrupting a process to complete a manual evaluation is inappropriate. Communicating the trust state of the platform

to the User is carried out by the Trust Communicator (TC). The role of the TC is to collate information received from the TH and present in an easily understood format to the user.

The exact role of the Reputation Manager (RM) is still to be defined. However, we already recognise that reputation is based on past events. Here, past events relates to how PII is handled by the Receiver. In addition to the negotiated contract, 'reports' will be issued by the Receiver indicating how either all of the PII or specific items that form the PII (e.g. name, address) have been handled. (The level of granularity, i.e. single PII item of complete PII package, that will be reported on is still to be agreed. Considerations include usefulness for determining trust and resolving disputes, User information overload and system performance.) Similarly these reports can be issued at various points during the time that the Receiver is in possession of the PII, indicating to the User the current status of the information. These ongoing confirmations provide assurance to the User that their PII is being managed as expected and further enhance the feeling of trust. Communicating this trust information to the User will be the responsibility of the TC. A point worth bearing in mind at this stage is that this information is essentially dynamic and could cause the User to reassess their trust in the Receiver at any point. It also becomes an event-driven model, similar to the function performed by the Obligation Manager (OM) and responsibility for managing these will fall in part to the Trust Real-time Monitor (TRM). A further point it considers is that as described so far, this is essentially a reputation service for the User. A similar requirement may exist for the Receiver, in which case indication that demonstrates fulfilment of the contract should be passed by the User to the Receiver.

An additional requirement is to provide the User with information about the trustworthiness of their local platform. Here the User is the owner of the platform and, from a hardware perspective, has control over its use. Users may however be concerned about the trustworthiness of other software installed locally, or about possible exposure should they lose their platform. Typically the User chooses to store personal information on the personal platform and uses a TPM to provide 'trusted storage' (i.e. where access to personal information is controlled by a specific hardware security device, the TPM). Access to the TPM to perform these functions is exactly the same as for an interaction with a remote entity, i.e. a local application routes requests through the AI and IDM to the TM.

The PCC requires access to the TM in order to determine compliance with policy, particularly where the policy relates specifically to trustworthiness. Just like the remote user, the PCC will hold a list of requirements and will look to the TM to indicate current status. The call that the PCC makes is exactly the same as the call made by the IDM. It is important to note that a User's first interaction with a remote platform is to determine its trustworthiness. This information gathering phase operates independently of any subsequent ID management processes, and once the user has determined the

acceptability of the trust status, the trust measurement phase effectively ends. However, there are opportunities for the user to re-examine their trust in a platform. For example, a platform that offers an obligation manager but does not provide deletion of credit card details may be considered untrustworthy. The fact would not normally be discovered until after the overall trustworthiness of the platform is determined during the initial interaction. Similarly, a change in platform status can affect trust. An attack on the TPM causing it to erase its keys should force a re-assessment. The responsibility for monitoring the on-going trust state of the platform falls to the Trust Real-time Monitor (TRM). The communication of information that describes the processing the PII has undergone could leak PII or imply its value in some other way. The IC (specifically the RDD and the LC) will be responsible for deciding how this communication takes place to minimise exposure.

17.6.1 Trust Handler (TH)

Within the TM it is the responsibility of the TH to gather information that describes the trustworthiness of a platform. The TH does so by 1) stating what trust technologies are supported and 2) indicating (where appropriate) the current status of trust technologies. For example, the TH might state that a Reputation Management module is present (supported) and that a TPM is installed, endorsed and running.

The TH is also responsible for providing access to the lower-level functions that these technologies support, e.g. TPM key management, which it does through the TW.

The TH presents information about the system by providing reputation information. Reputation information is derived from the Reputation Management module.

17.6.2 Trust Real-time Monitor (TRM)

The Trust Real-time Monitor (TRM) monitors the status of the platform trust indicators, e.g. the state of the TPM, on an on-going basis. Any discrepancies between the agreed trust profile for the platform and the measured value is alerted to the TH in the form of an event raised by the TRM.

17.6.3 Platform Trust Status (PTS)

The sole purpose of this module is to provide an interface between the TH and the system operating components (hardware and software). The module responds to a request for the status of a particular component by returning an appropriate value. For example, the query may concern the type and patch status of the operating system. Here the response would include the version number and latest patch ID.
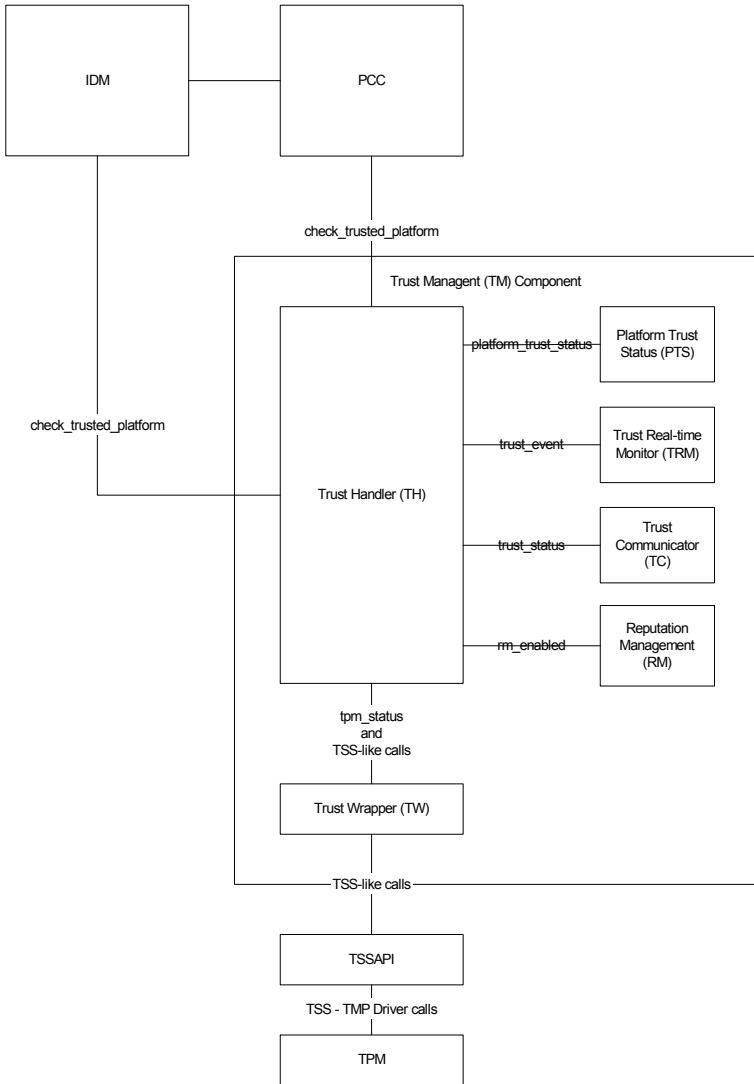
Fig. 17.3 PTM

17.6.4 Trust Communicator (TC)

The Trust Communicator (TC) is responsible for providing an easily under-
stood summary of the trust status of the platform to the user. This involves
gathering information from the remote and local THs, and translating this in-
formation into a human-readable format (probably icon-based). Note that the

TC is not responsible for the actual display (or rendering) of the trust status. Display is handled by the application (via the local IDM). The implication of this is that THs must be able to communicate with each other. (An alternative would be for a User to rely on a remote TC, which is feasible so long as the User trusts the remote TC and the communications path.) A further point to note is that the aggregation process that the TC performs should not be confused with the Trust Aggregator that forms part of the PCC (Services-side) and which provides an aggregated view of the trust state of several service-side platforms. In fact, the output of the TC could be considered an input to this module. Clearly, for this to occur the TC module must be present in the Services-side TM component.

17.6.5 Reputation Manager (RM)

The Reputation Management module is undefined beyond recognition that past performance can be an indication of future trustworthiness. An enquiry about a platform's trustworthiness should include an assessment of historic data, collect through pervious interaction with either a specific or sub-set of users, as an indication of the platform's willingness to fulfil any claims and obligations set. Exactly how this will be achieved is still to be determined, but for now one firm requirement is for the platform to announce the presence (or not) of reputation monitoring functionality. This will be initiated when the TM receives a request from the User (via the IDM). The TH is responsible for determining the presence of the RM and including this functionality in the list of supported trust mechanisms. All remote systems will possess an RM, but for now it will provide no functionality and announce itself as 'not enabled'.

17.6.6 Trust Wrapper (TW)

The Trust Wrapper provides the first level of abstraction from the hardware TPM (Trusted Platform Module).

17.7 Reputation Management

17.7.1 Objective Reputation Assessment

Reputation systems and auditing solutions are available and can provide opinions and ratings of service providers. However these assessments are usually very subjective, and not necessarily related to specific needs and requirements dictated by a user. In the absence of any other approach, these solutions can indeed be used to help users make a more 'informed' decision when engaging with another party.

However, these solutions do not easily handle 'follow-up' aspects, i.e., how users' expectations have actually been fulfilled. These expectations are specific for each user: people forget about promises and preferences. Their judgments could be emotional and very subjective. Current reputation systems mainly provide subjective, often only aggregated, ratings and opinions – they might not fit individual, specific needs.

17.7.2   Privacy Preferences and Privacy Obligations

We refer to two key concepts:

- **User preferences:** these are preferences and constraints on how users' personal data should be handled. This could include, for example, preferences on allowed purposes, data disclosure to third parties, data deletion date, notification preferences, etc.;
- **Obligations:** these are expectations and duties that an enterprise has to fulfil, on specific pieces of data [4]. These obligations are dictated by (and derived from) users' preferences, privacy policies (e.g. data protection laws), legislation (e.g. data retention laws) and internal guidelines.

Allowing individuals to express how they want their information handled is a key component of the design of our solution. The preferences we have chosen to use in our research to date are simply:

- **Delete:** delete my information at some time in the future. The options are: after the current transaction; in 30 days; in 6 months; keep forever;
- **Share:** share my data only with the organisation I specify. The options are: don't share; share with marketing; share with carefully selected partners; share with all;
- **Notify:** notify me at the time my information is going to be deleted or shared to another party. Our choice of preferences is based on evidence we gathered from our related Trustguide and Trustguide2 research projects.

These projects, which involved us engaging with citizen in order to understand why services and technology are not trusted, showed us that these few preferences are well aligned with what citizens demanded, and could be implemented objectively as obligations.

17.8   Conclusions

In this chapter we have given an assessment of how specific technological computer mechanisms can enhance trust.

18

Further Privacy Mechanisms

Anas Abou El Kalam[1], Carlos Aguilar Melchor[1], Stefan Berthold[2],
Jan Camenisch[3], Sebastian Clauß[2], Yves Deswarte[1], Markulf Kohlweiss[4],
Andriy Panchenko[5], Lexi Pimenidis[5], and Matthieu Roy[1]

[1] LAAS-CNRS
[2] TU Dresden
[3] IBM Research
[4] KU Leuven
[5] RWTH Aachen

18.1 Privacy Measures

In general, designing reasonable metrics for privacy quantification is an approach of several disciplines. This section focuses on technical and formal metrics. They can be distinguished depending on purposes or use-cases, available data, and the way results can be interpreted.

The purpose of a privacy metric for protocols or communication systems is to measure the degree of privacy which it can provide to its users. A purpose of a privacy metric for an individual user is to inform her about the privacy she may actually expect with respect to her situation, that is, for instance, her previous actions.

Data used for the privacy metrics can be available as persistent data, maybe organized in a database. In many cases, organizations carry out surveys and cut off names and addresses in order to call this an anonymous survey. However, if anonymity is only understood as cutting off the name and address, it is hard to decide in general whether the remaining attributes are sufficient to re-identify individuals or not. Privacy metrics assist in

assessing the significance of single attributes (or combinations of attributes) with respect to re-identification.

Another possible data source for privacy metrics can be a loose set of observations. A set of observations covers actions or events which occurred in a communication system over time. In contrast to what we called a persistent data source, these observations do not necessarily need to be complete or related to each other. However, the more complete the observations are, the more precisely privacy can be assessed and the less privacy will remain.

To assume an incomplete set of observations is probably more realistic, since even census data lacks details in many cases. However, no matter whether complete or not, errors may occur in databases as well as in observations. All discussed metrics in this section do not take any probability of errors into account.

The results of the metrics can be distinguished into possibilistic and probabilistic measures, and additionally in worst-case and average-case approaches. Worst-case approaches provide a metric for the least anonymity which a user may expect from the system. However, that can be too strict: In fact, a system which provides no anonymity in the worst case may work well in the majority of other cases. In this case, average-case anonymity metrics are better suited than worst-case metrics. This is particularly the case, if worst cases are rare or not relevant with respect to the targeted use-case. If, however, a system provides sufficient anonymity on average, but fails in relevant situations, worst-case anonymity metrics are best suited, indeed.

Possibilistic measures deal with anonymity sets directly. If subjects belong to the set, they are considered to be (definitive) anonymous, otherwise, they are not. The larger the set appears for an adversary, the stronger is the anonymity of subjects within the set. However, a great disadvantage is that possibilistic measures restrict the model to exactly one view of the world. Probabilistic measures, in contrast, deal with entropies, which are borrowed from information theory. The entropy of an observed attribute value yields the degree of information which an adversary is able to gain from his observation with respect to the adversary's prior knowledge. Thus, the entropy can also be used to estimate the size of the anonymity set which remains after the adversary's observation.

In the course of different traditions, several approaches have been developed for privacy metrics. In Section 18.1.1, we describe one of the main important approaches in the field of formal methods. In Section 18.1.2, we discuss work which has been motivated by surveys and statistical databases. In Section 18.1.3, we survey work which has been motivated by data-flow analysis in networks. In Section 18.1.4, we outline generalizations of the previous approaches and refer to work which is state of the art at the time of writing.

18.1.1   Formal Methods

Formal methods have been applied to privacy problems in two ways, (a) by using epistemic logics for reasoning about anonymity, cf. [SS99], or (b) by using process calculi for specifying the system behaviour, cf. [SS96]. Both approaches work well in their domain. Epistemic logics are well suited to state even complex anonymity properties. They lack, however, a trivial way to formalize system behavior. Process calculi are, in contrast, well suited to formalize processes and thus system behavior. However, stating anonymity properties in process calculi is difficult.

Hughes and Shmatikov propose a technique [HS04] which combines the advantages of both approaches. Their possibilistic metric consists of several layers, (i) the topmost layer providing anonymity properties on top of (ii) hiding properties on protocol graphs on top of (iii) the opaqueness of function views which are derived from (iv) function theory.

We explore this approach bottom-up. In Section 18.1.1.1, we briefly introduce function views as the fundamental concept behind the approach in [HS04]. In Section 18.1.1.2, we briefly introduce the notion of protocol graphs. In Section 18.1.1.3, we use an excerpt of anonymity properties specified in [HS04] to show how function views on protocol graphs correlate to privacy problems. In Section 18.1.1.4, we discuss related work.

18.1.1.1   Function Views

For specification of anonymity properties, Hughes and Shmatikov utilize the concept of function views [HS04]. Supposed, the capability of an adversary to obtain data is modeled by functions. For instance, a function $\mathsf{s} : M \rightarrow A$ assigns a sender (from a set of subjects A) to a conversation (from a set of conversations M). Then, anonymity properties can be expressed in a straightforward manner by restricting the adversary's knowledge about function s. Hughes and Shmatikov point out that it is sufficient for information-hiding to restrict three properties of functions, that is the graph, the image, and the kernel. These restrictions determine the view which an adversary has on the function.

The graph of an arbitrary function $f : \mathcal{A} \rightarrow \mathcal{B}$ is the corresponding relation graph $f \subseteq \mathcal{A} \times \mathcal{B}$ which consists right of those tuples (a, b) for which $f(a) = b$ holds with $a \in \mathcal{A}$ and $b \in \mathcal{B}$. Formally, we define the graph of function f as follows:

$$\operatorname{graph} f = \big\{(a, b) \mid f(a) = b\big\}$$

For instance, for

$$f(x) = \begin{cases} 1 & \text{for } x = 1 \\ 3 & \text{for } x = 2 \\ 3 & \text{for } x = 3 \end{cases}$$

we achieve graph $f = \{(1, 1), (2, 3), (3, 3)\}$.

The image of a function $f : \mathcal{A} \to \mathcal{B}$ consists of those elements $b \in \mathcal{B}$ for which there is an $a \in \mathcal{A}$ such that $f(a) = b$. Note that there may be elements $b' \in \mathcal{B}$ for which there is no $a \in \mathcal{A}$ that satisfies $f(a) = b'$. Formally, we denote

$$\operatorname{im} f = \{ f(a) \mid a \in \mathcal{A} \}$$

Continuing the example, we achieve $\operatorname{im} f = \{1, 3\}$.

The kernel of a function $f : \mathcal{A} \to \mathcal{B}$ consists of equivalence classes. These equivalence classes consists of all elements $a \in \mathcal{A}$ for which f maps to one and the same $b \in \mathcal{B}$. That is,

$$\langle a, a' \rangle \in \ker f \iff f(a) = f(a') \qquad \text{with } a, a' \in \mathcal{A}$$

Continuing the example, we achieve $\ker f = \{\{1\}, \{2, 3\}\}$.

A function view can then be denoted as a triple $\langle F, I, K \rangle$ where $F \subseteq \mathcal{A} \times \mathcal{B}$ describes the knowledge about the graph, $I \subseteq \mathcal{B}$ describes the knowledge about the image, and K is the equivalence relation on $\mathcal{A}$ which describes the knowledge about the kernel of f. In order to let the view $\langle F, I, K \rangle$ be restrictive with respect to f, the following constraints need to be satisfied:

- $F \supseteq \operatorname{graph} f$, that is the graph of the view leads to more uncertainty about the actual relation between inputs and outputs of f (or the uncertainty remains the same, if $F = \operatorname{graph} f$ holds). F can also be understood as approximation of f: $F(a) = \{b \in \mathcal{B} \mid (a, b) \in F\}$.
- $I \subseteq \operatorname{im} f$, that is the image of the view supports less unequal outcomes of f (or the same outcomes, if $I = \operatorname{im} f$ holds).
- $K \subseteq \ker f$, that is the kernel of the view is still a sound part of the kernel of f, however, the outcome of f depends on less equivalence classes of input values (or the outcome depends on the same equivalence classes, if $K = \ker f$ holds).

18.1.1.2 Protocol Graphs

The actual protocols are denoted as graphs $C = (\mathsf{s}_C, \mathsf{r}_C, \tau_C)$ over sets M, A, and T. That is, a protocol graph is a colored multigraph over a set A of *agents*, a set M of *abstract conversations*, and a set T of *relationship types*. Additionally, there are the labeling functions $\mathsf{s} : M \to A$ which maps conversations to the sender, $\mathsf{r} : M \to A$ which maps conversations to the recipient, and $\tau : A \times A \to T$ which maps a pair of agents to their relationship type. The functions s and r come from the definition of a multigraph $(A, M, \mathsf{s}, \mathsf{r})$, a directed graph with multiple edges between every pair of vertices, where s maps edges in M to their source vertex in A and r maps edges in M to their target vertex in A. The type of an edge can be understood as color of the graph. A colored multigraph (G, T, τ) consists of a multigraph $G = (A, M, \mathsf{s}, \mathsf{r})$, a set of colors T, and the type function τ which maps every pair of vertices to a color.

Protocol graphs $C = (\mathsf{s}_C, \mathsf{r}_C, \tau_C)$ can be easily denoted in process calculi. Suppose, a process calculus contains terms $\pi_{\mathcal{P}}(a, b, \tau, m)$ which represent a

conversation between two agents a and b, with m as a unique identifier of this conversation. A concurrent execution of k conversations can be denoted as P.

$$P = \pi_{\mathcal{P}}(a_1, b_1, \tau, m_1) \mid \cdots \mid \pi_{\mathcal{P}}(a_k, b_k, \tau, m_k)$$

The protocol graph C can be denoted as $P(C)$, accordingly.

$$P(C) = \pi_{\mathcal{P}}\big(\mathsf{s}_C(m_1), \mathsf{r}_C(m_1), \tau\big) \mid \cdots \mid \pi_{\mathcal{P}}\big(\mathsf{s}_C(m_k), \mathsf{r}_C(m_k), \tau\big) \quad \text{with } m_i \in M$$

18.1.1.3 Hiding and Anonymity Properties

Hiding properties can be defined by means of the opaqueness of the labeling functions s, r, and τ on protocol graphs and their compositions. In [HS04], eight (composed) functions are derived from the labeling functions:

$$\mathsf{s} : M \to A$$
$$\mathsf{r} : M \to A$$
$$\langle \mathsf{s}, \mathsf{r} \rangle : M \to A \times A$$
$$\tau : A \times A \to T$$
$$\tau^a : A \to T$$
$$\tau^a \circ \mathsf{s} : M \to T$$
$$\tau^a \circ \mathsf{r} : M \to T$$
$$\tau \circ \langle \mathsf{s}, \mathsf{r} \rangle : M \to T$$

Thereby, $\tau^a(b) = \tau(a, b)$. Hughes and Shmatikov distinguish in five kinds of opaqueness (or hiding properties), (i) k-value opaqueness, (ii) Z-value opaqueness, (iii) absolute value opaqueness, (iv) image opaqueness, and (v) kernel opaqueness. The functions and the kinds of opaqueness span a space of combinations, that is the space of anonymity properties which can be proven in this formalism. We focus our explanations on the function $\mathsf{s} : M \to A$.

As to (i), a function $\mathsf{s} : M \to A$ is k-value opaque for a view $\langle F, I, K \rangle$ and $k \geq 2$, if there are for each $m \in M$ at least k different $a_1, \ldots, a_k \in A$ such that $(m, a_i) \in F$ holds for each $i = 1, \ldots, k$. That is, there are at least k different output values valid for each input of function s. That can be considered as sender k-anonymity for the sender function s. The adversary could only discern senders up to a lineup of size k, cf. [HS04].

As to (ii), a function $\mathsf{s} : M \to A$ is Z-value opaque for a view $\langle F, I, K \rangle$ and some $Z \subseteq A$, if Z is a subset of the output set $F(m)$ for each input $m \in M$. That is, $\mathsf{s}(m) = a$ holds for each $a \in Z$ and each $m \in M$.

As to (iii), a function $\mathsf{s} : M \to A$ is absolute value opaque for a view $\langle F, I, K \rangle$, if it is Z-value opaque for $Z = A$. That is, every output $a \in A$ is a possibility for each input $m \in M$. That can be considered as absolute sender anonymity. That is, each agent would appear as plausible to be the sender in a conversation as any other agent.

As to (iv), a function $\mathsf{s} : M \to A$ is image opaque for a view $\langle F, I, K \rangle$, if $I = \varnothing$. That is, it cannot be asserted that any $a \in A$ is a valid output of function s. This would be required for unobservability. A slightly less strict flavor of image opaqueness is image value opaqueness where I might be known by the adversary, but not the actual image value $a \in A$ of s for a given $m \in M$. This could be understood as session-level sender anonymity[HS04].

As to (v), a function $\mathsf{s} : M \to A$ is kernel opaque for a view $\langle F, I, K \rangle$, if $K = \varnothing$. That is, it cannot be asserted that there is any $m' \neq m \in M$ for which holds $\mathsf{s}(m) = \mathsf{s}(m')$ or that there is any input value $m' \in M$ different from a given $m \in M$ which is mapped to the same output value $\mathsf{s}(m)$, respectively.

18.1.1.4   Related Work

Hughes and Shmatikov stress that function views can be used to mediate between system specifications which are commonly defined in process algebras and anonymity property specifications which are commonly defined in some logic. In a follow-up paper, Halpern and O'Neill [HO03] address the topic of mediation between system specifications and anonymity property specifications. They show that, though elegant and useful, function views are not necessary for mediation, since all the specification can be done by semantic characterizations.

18.1.2   Persistent Data and Statistical Databases

The question of privacy in databases of personal data records was tackled in large-scale when it came to the census discussion in (Western) Germany during the 1980s. Fischer-Hübner [FH87, FH01] points out that such data records consist of three kinds of data, that is *identity data*, *demographic data*, and *analysis data*. With identity data, it is possible to identify distinct persons, and thus, this data is obviously privacy-relevant. This could be, for instance, name or address. However, Fischer-Hübner also shows that the common assumption at that time, that truncating identity data would lead to anonymous data records, does not hold. It is rather obvious that combinations of demographic data, such as sex, nationality, education, religion, and marital status, can be used to re-identify people. Therefore, these items are also privacy-relevant.

There have been several complementary metrics [Rub93, BKP90], one of the most famous in the privacy-community has been proposed by Sweeney. In [Swe02], she points out that classical access control approaches fail to protect against data disclosure. This particularly is the case, if protected data is not subject of the release process, but results from the derivation of legitimately released data. Therefore, at least unambiguous relations between released data and supplementary knowledge must be avoided.

The actual threat which arises from database contents depends on the recorded attributes and the frequency distribution of their values. Fischer-Hübner proposes a probabilistic metric for assessing the *risk of re-identification*,

whereas the *k-anonymity* metric proposed by Sweeney is possibilistic. Therefore, *k-anonymity* is only applicable for worst-case considerations, whereas Fischer-Hübner's metric yields average-case results.

Both metrics require quite strong assumptions. The data-holder (or whoever wants to assess the threats of re-identification) has to determine the quasi-identifier[1] properly, in any case. The assumption is that she does so. Furthermore this person needs access to all databases which could be used for identification. In case of public databases, this requirement is satisfied. If the adversary uses supplementary data which is not publicly available, then the quasi-identifier cannot be chosen appropriately.

18.1.2.1   Risk of Re-identification

The metric of Fischer-Hübner [FH87, FH01] can be understood as a metric of uniqueness of attribute values (or combinations) within a database. Her metric bases on Shannon entropies [Sha48].

Supposed, there is a database table with n records. Each record consists of values for a set of discrete attributes, including $X_1, \ldots, X_m$.

Fischer-Hübner defines the risk of re-identification $r(X_1, \ldots, X_m)$ as the ratio between the *average number of value combinations* $n_{vc}(X_1, \ldots, X_m)$ that can be used for re-identification and n.

$$r(X_1, \ldots, X_m) := \min\left(1, \frac{n_{vc}(X_1, \ldots, X_m)}{n}\right) \tag{18.1}$$

We need to enforce 1 as the upper bound, since the ratio can in fact be greater than 1. This happens, if the number of such value combinations is greater than the number of records. However, a risk greater than one has no useful interpretation.

The average number of value combinations which can be used for re-identification is defined by means of the entropy $H(X_1, \ldots, X_m)$ of all involved attributes $X_1, \ldots, X_m$. That entropy yields the information which can be obtained from the corresponding values to these attributes. The entropy's dimension is bit, thus, the average number of value combinations which contributes information is $2^{H(X_1, \ldots, X_m)}$, that is the number of states of a single bit, to the power of the entropy.

$$n_{vc}(X_1, \ldots, X_m) := 2^{H(X_1, \ldots, X_m)} \tag{18.2}$$

The greater the entropy is, indeed, the greater is the number of value combinations which can be used for re-identification. In fact, the entropy is the greater the more the frequency distribution of all involved attribute values converges to uniform distribution. Furthermore, it is the smaller the

---

[1] The *quasi-identifier* is a set of database attributes which unambiguously identifies a subject within the database.

more uneven this frequency distribution is. In addition, the entropy depends also on the number of involved attributes. The greater this number is, the greater is the entropy and vice versa.

Strictly speaking, the joint entropy $H(X_1, \ldots, X_m)$ is defined as a sum of conditional entropies. We do not elaborate the details in this section, however, more detailed background can be found in [Sha48]. By means of conditional entropies, Fischer-Hübner's metric can also take dependencies between different attributes into account.

18.1.2.2   k-Anonymity

Sweeney [Swe02] proposes a possibilistic metric. It can be used to assess threats of re-identification which arise from linking attributes which are shared between different databases.

Supposed, two database tables overlap in a subset of a quasi-identifier, then we can count the occurrences of records with the same values in this attribute subset. The actual measure which is provided by k-anonymity is k which denotes the smallest count of these occurrences.

The records from both databases cannot unambiguously be linked as long as k is greater than 1. The reverse, however, does not generally hold, since $k = 1$ only states that at least one record can be linked. In particular, it makes no difference with respect to k, if just one or many more records can be linked. The greater k is, however, the stronger is the anonymity assumed to be.

18.1.3   Data-Flow in Networks

There are several different approaches for systems and protocols which were meant to preserve the user's anonymity, for instance, the DC net [Cha88], mix-based approaches [Cha81, DMS04], or Crowds [RR98]. The efficiency of these approaches with respect to required resources can be assessed by means of traditional analysis methods. Assessing the anonymity which they provide, however, turned out to be a more severe problem. In this section, we focus on metrics which tackle this topic.

Díaz et al. proposed a metric [DSCP02] which assesses the sender anonymity that can be provided by a communication system. A similar approach has independently been proposed by Serjantov and Danezis [SD02, Dan03] at around the same time. The difference between both metrics is mainly that Díaz et al. normalize the entropy, whereas Serjantov and Danezis use the entropy measure without normalization. In [Dan03], Danezis discusses the pros and cons of normalization with respect to these measurements. His conclusion is basically that, by normalization, important information about the measured anonymity gets lost, particularly the average size of the corresponding anonymity set. However, Díaz et al. advocate a quality measurement which is independent from anonymity set size and only relies on the distribution

of probabilities. That is, the probabilities of users for being the sender of a particular message.

The foundation of both metrics is Shannon's information theory [Sha48]. Supposed that adversaries are able to carry out observations and assign corresponding probabilities to possible senders of a message. There are various kinds of observations which an adversary may use, for instance results from traffic analysis, timing attacks, message length attacks, or generally information leaks of the communication system. By means of assigning probabilities, adversaries are able to distinguish possible senders of a message much better than by assigning them to anonymity sets. Entropy, that is the information which is contained in a given distribution of probabilities, is then used to assess the information which the adversary was able to obtain. Or, from the point of view of a user, entropy is used to assess the anonymity which the user was able to preserve with respect to the adversary's observation.

Díaz et al. define the information leak of the adversary's attack as the difference between maximum entropy of the system and the actual entropy of the system after the adversary's observation. Thus, by denoting the maximum entropy as H_M—note that this must not be confused with max-entropy—which is possible[2] and the entropy after an observation as $H(X)$, the information which the adversary has learned can be assessed by $H_M - H(X)$. Here, X is a discrete random variable with probability mass function $p_i = \Pr(X = i)$, where i is an index over all users in the system. The entropies $H(X)$ and H_M can be calculated as shown in Equation 18.3, the latter one by means of the number of all users in the system n.

$$H_M = \log_2(n) \qquad\qquad H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \qquad (18.3)$$

The *degree of anonymity*, denoted as d, is then defined as the difference between the state of perfect anonymity and the adversary's gain of information. As mentioned, this degree is normalized with respect to H_M:

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M} \qquad (18.4)$$

This degree is a value between 0 and 1, where 0 denotes no preserved anonymity and 1 denotes perfect anonymity with respect to the system. That is, the adversary is able to identify the user as sender of the message, in case of $d = 0$. The other extreme value $d = 1$ would mean that the adversary is just able to guess the sender, since all users appear evenly suspicious for having sent the message. This case is similar to an anonymity set which contains basically all users. And otherwise, that is d is neither 0 nor 1, it holds, the greater d is the stronger is the average anonymity.

---

[2] This entropy refers to the maximum which is possible within the given system. It is very unlikely that this optimum will ever be reached in practice with real persons as users.

With this normalization, it is possible to assess arbitrary communication systems with respect to a given lower bound of anonymity. Díaz et al. point out, however, that such a lower bound depends very much on system requirements and can hardly be suggested, generally.

Serjantov and Danezis [SD02, Dan03] use $H(X)$ without any normalization to assess the anonymity. This yields the average set size of a corresponding anonymity set and, therefore, a measure about the actual effort which an adversary has to take into account for identifying a user as sender of a message. This average size k of the anonymity set can be calculated by means of the dimension of the entropy $H(X)$ which is bit:

$$k = 2^{H(X)} \tag{18.5}$$

These metrics can easily be adapted for recipient anonymity or any other action.

18.1.4   Generalizations

Both measurements which have been described in the previous section are useful to quantify the effort of an adversary to compromise all messages, that is to assign the messages to senders. Tóth et al. refer to this quantification as global measure [THV04] and point out that it is of little use for users to quantify their particular anonymity. They refer to the latter quantification as local aspect of anonymity and prove that different probability distributions which provide very different local anonymity lead to the same level of anonymity with respect to global measurements. Furthermore, they show that for a given degree of anonymity there is always a corresponding probability distribution which is not desirable for all users. Thus, Tóth et al. conclude, global measures do not provide a quantification of anonymity with respect to local aspects.

18.1.4.1   Local Anonymity

In order to overcome the shortcomings, they propose [THV04] an upper bound Θ and suppose that an adversary is successful, if she can assign a message to a user with probability greater than this upper bound. Thus, a system provides sender anonymity as long as for all received messages β and all senders s holds that the probability $p_{\beta,s}$ for s being the sender of β is lower or equal Θ, formally

$$\forall \beta. \forall s. \big(p_{\beta,s} \leq \Theta \big) \tag{18.6}$$

Dually, this can be formalized for recipient anonymity.

This is a generalization of global measures, since the (global) degree of anonymity d can be assessed as well as $H(X)$ by using Θ:

$$H(X) \geq -\log_2 \Theta \tag{18.7}$$

$$d \geq -\log_n \Theta \qquad \text{where } n \text{ is the number of senders} \tag{18.8}$$

18.1.4.2 Towards Arbitrary Attributes

The ideas of this section mainly reflects the current work of Sebastian Clauß which is elaborated within a greater context in his PhD thesis [Cla07b].

Modelling the Observer's Knowledge Base

By observing actions, an observer gets a limited insight into user's personal information (hence we address it as a set of attributes) and into relations between different attribute values. The observer can collect this information, and may conduct any desired statistical analysis on them. With a growing number of observations, the information on the relative frequency of the digital identities gets more exact[3]. The knowledge of an adversary which he gained by observations in form of the *observer state* is defined as follows:

Definition 7 (Observer State). *Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be n attributes where each attribute comprises a set of attribute values. The* State $Z^{\mathcal{X}}$ *of an observer $\mathcal{X}$ is a triple $(\mathcal{I}, h, g)$, where:*

- $\mathcal{I}$ *is the set of all possible digital identities.*

$$\mathcal{I} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$$

- $h : \mathcal{I} \to \mathbb{R}$ *is a function, which assigns a relative frequency to each digital identity.*

$$\forall i \in \mathcal{I}.\ 0 \le h(i) \le 1$$

- g *is the number of observations leading to this* state.
- *the sum of all relative frequencies is 1.*

$$\sum_{(\mathcal{I})} h(i) = 1$$

The value $h(i)$ denotes the relative frequency of the adversary discerning identity i from all other identities in $\mathcal{I}$. When the adversary observes a user's action, the relative frequency of the identities matching the observation (i.e., the suspects with respect to the observation) is increased, whereas the probability of all other identities is decreased.

Definition 8 (Observation). *An* observation *is a (possibly incomplete) bundle of attribute values. Such a bundle contains at most one value per attribute. The set $\mathcal{B}$ of all possible* observations *is the cross product of all attributes with an additional element "not observed" $\perp$.*

$$\mathcal{B} = (\mathcal{A}_1 \cup \{\perp\}) \times (\mathcal{A}_2 \cup \{\perp\}) \times \cdots \times (\mathcal{A}_n \cup \{\perp\})$$

---

[3] "Exact" here means exactness with respect to the observation. Observations may nevertheless yield incorrect information.

Intuitively, this means that a user discloses attribute values whenever she performs any action. The observer *observes* this values and, by time, gets a more refined view on the digital identities and thus on the users.

Within the set of all possible digital identities, an observer can separate suspect digital identities with respect to an observation from non-suspect digital identities. The set of *suspects* related to an observation can be defined as follows:

Definition 9 (Suspects). *The set of* suspects $\mathcal{V}_b$ *related to an observation* $b = (x_1, .., x_n)$ *contains all digital identities* $i = (x'_1, .., x'_n)$, *whose attribute values are either equal to attribute values of b, that is* $x_k = x'_k$, *or are not contained in b, that is* $x_k = \bot$.[4]

$$\mathcal{V}_b = \mathcal{V}_{(x_1,\ldots,x_n)} = \left\{ (x'_1, \ldots, x'_n) \in \mathcal{I} \mid \forall k \in \{1, \ldots, n\}. x_k \in \{x'_k, \bot\} \right\} \quad (18.9)$$

As already mentioned, the observer *learns* by observations. The following definition formalises this learning process:

Definition 10 (Observer State Update). *Let* $b \in \mathcal{B}$ *be an observation and* $\mathcal{Z}$ *a set of observer states. An* observer state update $\delta : \mathcal{Z} \times \mathcal{B} \to \mathcal{Z}$ *constructs a new observer state from a given state and an observation. An observer state update does not change the set of identities $\mathcal{I}$ of the observer state.*

These definitions can be seen as a framework for formalising concrete observations and statistical analysis based on digital identities. In particular, the way observations are gathered can be chosen in applications according to the current adversary model.

Based on the previous definitions, a statistical observer model can be defined as follows:

Definition 11 (Statistical Observer Model). *A* statistical observer model *of an observer* $\mathcal{X}$ *comprises a set of digital identities $\mathcal{I}$, a set of observations* $\mathcal{B}$, *a set of observer states $\mathcal{Z}^{\mathcal{X}}$, and an update function δ which derives new observer states from previous states and observations.*

The statistical observer model specifies the observer's knowledge in form of statistics about digital identities together with a method for aggregating newly gained knowledge. This is an abstract definition, as it leaves open how the aggregation of new observations actually influences the relative frequencies of digital identities.

In order to aggregate knowledge about entities within the system, a concrete observer state update method needs to be defined. That is, given an

---

[4] The matching function "equality" used here is a simple example. This makes sense, if attribute values are discrete and not related to each other. If this is not the case, e.g., if measuring faults for actually continuous attribute values need to be taken into account, other matching functions should be used in order to reflect such properties of attributes appropriately.

observer state, it needs to be modelled how the relative frequencies of the digital identities change after an observation.

Thereby, the major goal is that, by observations, the frequency distributions of attribute values within the observer state shall converge to the actual probabilities of digital identities in the real world.

An example for an update function has been proposed by Clauß in [Cla06]. This update function lets the frequencies in the model converge to the probabilities in the real world as long as the observations are either complete or the attribute values of not observed attributes are all uniformly distributed.

18.1.4.3 Unlinkability

Steinbrecher and Köpsell [SK03] propose another generalization which introduces the notion of unlinkability and also takes the local anonymity into account. Particularly, they point out how anonymity can be quantified in terms of unlinkability. In contrast to anonymity which is a property of subjects, unlinkability could be a property between arbitrary items, that is subjects, actions, events, etc. Unlinkability of items with respect to an observation of the adversary holds, if the items are not more and not less related for the adversary before and after his observation. Thus, if a sender was anonymous before an adversary's observation and unlinkability holds with respect to the sender and her message, then the sender remains anonymous after the observation.

Steinbrecher and Köpsell model the relation between items as equivalence relation on items. The adversary is supposed to know the items, but not to know the equivalence relation. However, the adversary may eventually gain knowledge about the equivalence relation. This gain in knowledge is modelled by a change in the probabilities for each possible equivalence relation.

This approach is first applied to model unlinkability between two items and then successively to more complex issues. We denote the relation between two items a_i and a_j within a set[5] $\mathcal{A}$ as $a_i \sim_{r(\mathcal{A})} a_j$. Furthermore, we denote the probability which the adversary assigns to this relation as $\Pr(X = (a_i \sim_{r(\mathcal{A})} a_j))$ for a random variable X or, in short, $\Pr(a_i \sim_{r(\mathcal{A})} a_j)$. Accordingly, $\Pr(a_i \nsim_{r(\mathcal{A})} a_j)$ denotes the probability that the items a_i and a_j are not in relation.

The entropy $H(i, j) := H(X)$ can then be used as a measure for the degree of unlinkability $d(i, j)$ of the two items a_i and a_j.

$$
\begin{aligned}
d(i,j) &= H(i,j) \qquad\qquad\qquad\qquad\qquad\qquad (18.10)\\
&= -\Pr(a_i \sim_{r(\mathcal{A})} a_j) \cdot \log_2\big(\Pr(a_i \sim_{r(\mathcal{A})} a_j)\big)\\
&\quad -\Pr(a_i \nsim_{r(\mathcal{A})} a_j) \cdot \log_2\big(\Pr(a_i \nsim_{r(\mathcal{A})} a_j)\big)
\end{aligned}
$$

---

[5] This could be an anonymity set. However, if we would explicitly write about anonymity sets here, we would unnecessarily lose generality for the types of items and entirely stick to subjects, instead.

The degree $d(i,j)$ becomes 0, if the adversary is either certain of $a_i \sim_{r(\mathcal{A})} a_j$ or of $a_i \not\sim_{r(\mathcal{A})} a_j$. The degree becomes 1, if the adversary is completely uncertain of the relation between a_i and a_j, that is $\Pr(a_i \sim_{r(\mathcal{A})} a_j) = 0.5$ as well as $\Pr(a_i \not\sim_{r(\mathcal{A})} a_j) = 0.5$. The former case describes perfect linkability, whereas the latter case describes perfect unlinkability.

Similarly, the unlinkability of a set of items can be quantified. Let $A \subseteq \mathcal{A}$ be a subset of all items with $|A| > 2$ and $\sim_{r(A)}$ be an equivalence relation on A. The item set A denotes all items which an adversary observes and $\sim_{r(A)}$ denotes a guess of equivalence classes in A. Therefore, the probability for an adversary to succeed with linking is the probability for $\sim_{r(A)}$ being the same as $\sim_{r(\mathcal{A})}$, the actual equivalence relation, restricted to the elements of A, formally

$$\Pr\left(\sim_{r(A)} = \sim_{r(\mathcal{A})}\big|_A\right) \tag{18.11}$$

The degree of unlinkability with respect to A can then be calculated by means of the enumeration I_k of all possible equivalence relations on A and the entropy of the corresponding probability distribution.

$$d(A) = H(A)$$
$$= -\sum_{j \in I_k} \frac{1}{|I_k|} p_j \cdot \log_2 p_j \tag{18.12}$$
$$\text{where } p_j = \Pr\left(\sim_{r_j(A)} = \sim_{r(\mathcal{A})}\big|_A\right)$$

The degree $d(A)$ is 1, if the adversary is certain of one $\sim_{r_j(A)}$ being the same as $\sim_{r(\mathcal{A})}$ restricted to elements of A. The degree $d(A)$ is 0, if the adversary is completely uncertain about all $\sim_{r_j(A)}$, that is $\Pr(\sim_{r_j(A)} = \sim_{r(\mathcal{A})}|A) = 0.5$ for each $r_j \in I_k$.

Steinbrecher and Köpsell pointed out, however, that it is not sufficient to address unlinkability within one set only. In order to describe anonymity in terms of unlinkability, it is rather necessary to address unlinkability between different sets. This could be, for instance, a set of messages and a set of senders. Sender anonymity can then be described by unlinkability between senders and messages.

The definition of unlinkability between two different sets $\mathcal{A}$ and $\mathcal{B}$ is similar to unlinkability between two items within the same set. The equivalence relation between two items within the same set, however, has to be replaced by a relation $\sim_{r(\mathcal{A},\mathcal{B})}$ between items in $\mathcal{A}$ and $\mathcal{B}$. The relation $\sim_{r(\mathcal{A},\mathcal{B})}$ itself is no equivalence relation, however, equivalence relations $\sim_{r(\mathcal{A})}$ and $\sim_{r(\mathcal{B})}$ can be constructed by means of capturing all $a \in \mathcal{A}$ in equivalence classes of $\sim_{r(\mathcal{A})}$ which are related to the same $b \in \mathcal{B}$ (and vice versa for $\sim_{r(\mathcal{B})}$).

The degree of linkability of two items within different sets $d(a,b)$ can then also be reduced to entropy.

$$d(a,b) = H(a,b) \tag{18.13}$$
$$= -\Pr(a \sim_{r(\mathcal{A},\mathcal{B})} b) \cdot \log_2\big(\Pr(a \sim_{r(\mathcal{A},\mathcal{B})} b)\big)$$
$$- \Pr(a \nsim_{r(\mathcal{A},\mathcal{B})} b) \cdot \log_2\big(\Pr(a \nsim_{r(\mathcal{A},\mathcal{B})} b)\big)$$

18.1.4.4   Rényi Entropy

This section deals with the calculation of privacy parameters based on a given observer state. The former two parts of this section refer to the case that a user has exactly one digital identity. The latter part describes how calculations have to be done in case users may have multiple digital identities.

Quantifying Anonymity

Definition 12 (Shannon entropy[Sha48]). *Let b be an observation and $\mathcal{V}_b$ a set of suspects related to observation b. The Shannon entropy of b is the Shannon entropy of the suspects $\mathcal{V}_b$.*

$$H_\varnothing = - \sum_{(v \in \mathcal{V}_b)} \Pr(v|b) \log_2 \Pr(v|b) \tag{18.14}$$

$$\Pr(v|b) = \frac{\Pr(v \wedge (\bigvee_{(w \in \mathcal{V}_b)} w))}{\Pr(\bigvee_{(w \in \mathcal{V}_b)} w)} \tag{18.15}$$

Given a Shannon entropy $H_\varnothing$, $|\mathcal{S}| = 2^{H_\varnothing}$ denotes the corresponding size of a uniformly distributed anonymity set $\mathcal{S}$. The Shannon entropy $H_\varnothing$ specifies the average amount of information needed in addition to b in order to uniquely identify a digital identity.

Here we refer to the case that a user has only one digital identity, so that a measurement related to a digital identity can be seen synonymous to a measurement related to the user, who "owns" this digital identity. The Shannon entropy $H_\varnothing$ specifies the average amount of information needed in addition to b in order to uniquely identify a digital identity. In case of a user evaluating her anonymity, she usually knows her digital identity. So, it may be more useful for her to compute the amount of information needed to identify *her*, i.e., her digital identity. This so called *individual anonymity* can be computed as follows:

$$H(v) = - \log_2 \Pr(v|b) \tag{18.16}$$

From the viewpoint of each single user, *individual anonymity* is the most accurate probabilistic anonymity measure.

Example 5. An observer knows that within a given source of information the element A shows up with a probability of 0.5. If the observer is only interested in the occurrence of A (i.e. how anonymous A is), this is independent of the Shannon entropy of the information source. The anonymity measure of A

only depends on the probability of A's occurrence. On the other hand, the Shannon entropy also depends on the number of elements of the information source. So, even if A occurs with a probability of 0.5, the Shannon entropy can have an arbitrarily high value depending on number and distribution of the other elements of the information source. However, the amount of information needed to identify A remains the same, independent from the Shannon entropy of the information source.

It is also possible to specify a worst-case measure for anonymity [THV04]. This is the individual anonymity of the identity with the highest probability (also called *Min-entropy*):

$$H_{\text{Min}} = -\log_2 \max_{v \in \mathcal{V}_b}(\Pr(v|b)) \tag{18.17}$$

In [CS06], Clauß et al. discussed use of *Rényi entropy* as a more general metric for anonymity. Rényi entropy H_α, introduced by Rényi [Ren60], is defined as follows:

$$H_\alpha = \frac{1}{1-\alpha} \log_2 \sum_{(v \in \mathcal{V}_b)} \Pr(v|b)^\alpha \tag{18.18}$$

Besides the probability distribution given, Rényi entropy incorporates an additional parameter α. In Figure 18.1 the influence of α on Rényi entropy is shown. The more α grows the more the Rényi entropy converges to Min-entropy H_{Min}. On the other hand, the more α runs to zero the more H_α converges to Max-entropy $H_{\text{Max}} = \log_2 N$, where N is the number of elements of the probability distribution given[6]. Furthermore, if α runs to 1, Rényi entropy converges to Shannon entropy. The proofs of these facts are given in [CS06].

By adjusting the parameter α, it is possible to fade between worst-case anonymity, average case anonymity, and k-anonymity. Thus, for evaluating anonymity within a given system, the parameter α can be adapted according to certain characteristics of the system.

Quantifying Linkability of Actions

Regarding linkability, it is interesting for an attacker, to what extent it can be determined that actions have been done by the same user. More formally, there are two actions c_1 and c_2 which have been observed in the form of observations b_1 and b_2.

According to [SK03], linkability of items of interest can be measured regarding equivalence classes, for which (after observations) an adversary has partial knowledge about which items of interest belong to which class.

---

[6] H_{Max} directly corresponds to k-anonymity. It denotes the entropy of a source with k elements, thereby ignoring the probability distribution of the elements, i.e., assuming a uniform probability distribution.
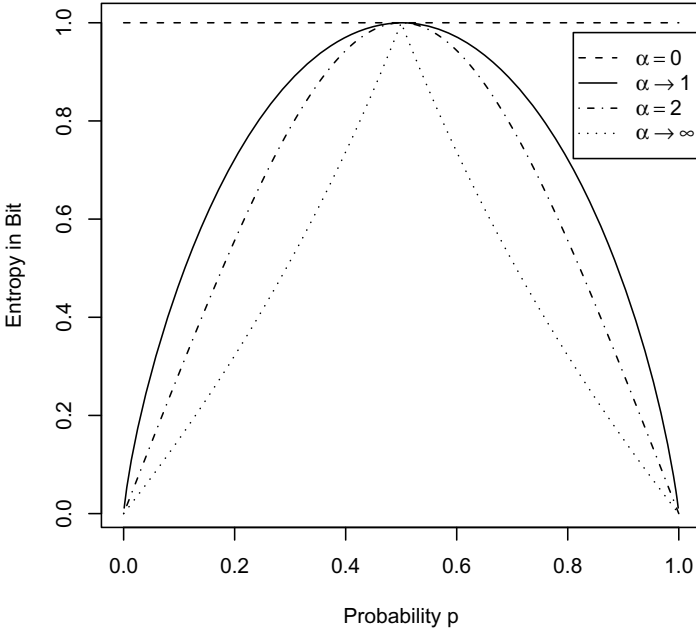
Fig. 18.1 Influence of parameter α on Rényi entropy of a source containing two elements with probabilities p and $1 - p$ resp.

Applied to the model used here, the equivalence classes are the digital identities. By an observation of an action, suspect digital identities can be determined corresponding to the observation of this action (see Definition 9), i.e. information about association of items of interest (actions) to equivalence classes (digital identities) is gained.

Regarding observations b_1 and b_2, the suspect sets are $\mathcal{V}_{b_1}$ resp. $\mathcal{V}_{b_2}$. Within a set of suspects, a digital identity has the probability $\Pr(v|b)$.

The probability p_r, that actions c_1 and c_2 belong to the same digital identities, can be computed as follows:

$$p_r = \sum_{(v \in \mathcal{V}_{b_1 \wedge b_2})} \Pr(v|b_1) \cdot \Pr(v|b_2)$$

Thereby, $\mathcal{V}_{b_1 \wedge b_2}$ denotes the set of digital identities, which are contained in both sets $\mathcal{V}_{b_1}$ and $\mathcal{V}_{b_2}$, i.e. which are suspects of both observations, b_1 and b_2. Consequently, the probability $p_{\neg r}$, that the actions c_1 and c_2 do not belong to the same digital identity is $1 - p_r$.

From probabilities p_r and $p_{\neg r}$ a degree of linkability d can be computed by using the Shannon entropy [SK03]:

$$d = H(p_r, p_{\neg r}) = -p_r \cdot \log_2 p_r - p_{\neg r} \cdot \log_2 p_{\neg r}$$

The events "actions c_1 and c_2 belong to the same digital identity" and "actions c_1 and c_2 do not belong to the same digital identity" are used as elements of a two-element source of information. The degree of linkability d is the Shannon entropy of this source. It specifies, how much an observer has learnt about the relation between c_1 and c_2 from observations $\mathcal{V}_{b_1}$ and $\mathcal{V}_{b_2}$.

The maximum degree of linkability, $d = 1$, means that the observer does not know anything about whether actions c_1 and c_2 belong to the same digital identity or not.

If $p_r > p_{\neg r}$, the degree denotes the certainty of the observer, that actions c_1 and c_2 *belong to the same digital identity*, otherwise it denotes the certainty of the observer that the actions do *not belong to the same digital identity.*

In case a user has only one digital identity, linkability related to a digital identity is the same as linkability related to a user. The next section deals with users having multiple digital identities.

Users with Multiple Digital Identities

In real life, a user will often not only have one digital identity, but lots of them. So, for example a user may have many different e-mail addresses, which she uses in different situations. Nevertheless also in this case, a user will be interested in *her* privacy, and not only in the privacy of one of her digital identities.

In order to calculate privacy parameters for users having multiple identities, we can first determine suspect digital identities as described for the different metrics in the previous sections. Now, in order to calculate measurements with respect to *users*, we need to group suspect digital identities belonging to the same user into *personal* digital identities. Thereby, grouping means that for each user the probability values of all digital identities belonging to this user are summed up.

After the probabilities of *personal* digital identities are determined, calculations of anonymity and linkability metrics can be done as described above, but based on probabilities of these *personal* digital identities.

18.2   Data Anonymization

18.2.1   Introduction

Data anonymization aims at providing privacy by cutting the link between personal data and the person or persons they refer to. In that case, anonymized data sets can be used for various purposes (medical research, demographic

statistics, customer and market analyses, etc.), without disclosing sensitive data that could endanger privacy of individuals. In practice, anonymization consists of removing all directly identifying data from the personal data sets. Examples of directly identifying data are civil identities (name, first name, etc.) or Social Security Numbers: with such identifiers, it is easy to retrieve a person from the related personal data. In most cases, a person can also be identified uniquely by other "nominative" data, such as a login name, a telephone number or an IP address (at least at precise times). So it is also necessary to remove them to anonymize personal data sets.

But this is not sufficient, and several questions need to be addressed before implementing a data anonymization process. In particular:

1. Which data are to be considered as nominative or personal?
2. In specific cases, which data should be collected, and which precise authorities and persons may have access to these data?
3. Which security mechanisms to use, in which cases and at which level?

Concerning the first question, there is a certain difference between nominative and personal data. In fact, even after anonymization (i.e., erasing directly identifying data, such as names or postal addresses), it is sometimes possible to identify a person by linking some of his/her data. For example, the age, gender and month of discharge from an hospital are sufficient to identify the patient in a limited population. Likewise, knowing two childbirth dates is sufficient to identify one woman in a sizeable population like that of France.

The response to the second question requires taking into account the purpose of use. For example, data collected for a global evaluation of the impact of disease prevention actions, would be different from those collected to establish a fine epidemiological surveillance of the HIV evolution.

Concerning the third question, we generally call on technical solutions (such as symmetric or public key encryption, one-way hash functions, access control mechanisms) and organizational solutions (such as security policy) to ensure privacy [MOV96] [Sch96]. For instance, encrypting transmitted data can provide a certain level of confidentiality; this can be useful, e.g., when transmitting medical data between test laboratories and physicians. In other cases, the aim is that patient's identity remains anonymous even if the receiver is legitimate (e.g., in scientific publications). In these cases, we can call on technical solutions such as a one-way hash function (e.g., SHA-2). Furthermore, if the aim is to have a permanent follow-up of certain diseases, it may be necessary to keep patient data anonymous, while keeping the possibility to crosscheck information belonging to the same patient but coming from different sources at different times.

Finally, it is sometimes desirable that certain authorities can crosscheck anonymized data with other anonymous data belonging to the same patient, or even, to re-identify the patients in some particular situations. For example, with the aim of refining epidemiological studies, correlations between several pathologies can necessitate crosschecking data previously collected for

different purposes on the same patient, or it can be vital to inform a patient of new therapy results.

Subsequently, we can conclude that the privacy needs can differ according to the purpose of the collected data. Addressing these issues, the remainder of this chapter is organized as follows:

In order to emphasize the notions presented above, Section 18.2.2 analyzes some anonymization examples in Europe and the USA. The aim is to progressively derive the requirements that should be satisfied by a suitable solution (Section 18.2.3). Afterwards, Sections 18.2.4 and 18.2.5 proposes a generic anonymization architecture and implementation that meets the privacy requirements previously identified. Finally, Section 18.2.6 presents future trends and we dress some conclusions concerning these topics in Section 18.2.7.

Note that thanks to its richness, the healthcare domain appears as the most demanding domain regarding privacy and anonymization, and thus our examples will be taken from this field.

18.2.2   Analysis of Some Anonymization Examples in Europe and the USA

18.2.2.1   Example of Anonymization in the United States

In the United States, the Social Security Administration uses a *Tricryption Engine* (TE) to protect medical data. The TE is a large encryption and automated key management system. It encrypts data with a per-call generated cryptographic key, encrypts the key and encrypts the link between the data and the key. The full process is the following [Vas05]:

- Sensitive data to be encrypted are selected by the user, and a request for encryption is sent to the TE.
- A randomly generated, symmetric session key is created; and a random Key ID is created. The session key is encrypted.
- The encrypted key and its Key ID are stored in a Key DB.
- The Key ID is encrypted, producing a Hidden Link.
- The personal data are encrypted, using the session key.
- The encrypted data and the Link are returned to the user.
- The encrypted data and the session key used to encrypt them are completely separated,
- both physically and logically, and the link between them is hidden.

However, as the details of this solution are not published, several question remains without answer:

1. As the tricryption engine has a complex architecture based on several separate modules (key databases, authentication, authorization; public key infrastructure; etc.), the interoperability of these modules is a serious practical problem. Indeed, how are these components managed, updated, interconnected?

2. The system contains several critical entities whose corruption endanger the whole system security. For example, what happens if the key used to encrypt the session keys is corrupted? The only solution is to decrypt all the key databases and to re-encrypt them with a new key; moreover, as the encrypted session keys are changed, the hidden links must also be changed, and so on.

3. Actually, the privacy needs (anonymization or pseudonymiszation or linkability or observability) are not clearly identified; it is thus difficult to judge if this solution is well-adapted. In fact, should we really encrypt all the data (that is the matter with this solution) or it is sufficient to only encrypt some selected data?

18.2.2.2 Example of Anonymization in France

In 1995, The Dijon University Hospital Center and other French health establishments implemented an anonymization protocol [QBAB98]. The aim is to ensure an irreversible transformation (i.e., anonymization) of a set of variables that identify an individual (last name, first name, date of birth, gender). This protocol transforms patient identities by using a one-way hash coding based on the *Standard Hash Algorithm* (SHA). In order to link all the information concerning the same patient, the anonymous code obtained is always the same for the given individual.

However, although mathematically irreversible, the hash computation does not guarantee information security and unlinkability. Indeed punctual, as well as dictionary attacks remain possible on the anonymous medical databases; e.g., by comparing hashed known identities with the code assigned to a certain patient. Let us take a simple example. Assume that *Bob*, a malicious person, has access to an anonymous medical database *AMDB*. In a punctual attack, *Bob* wants to know if *Alice* has AIDS. By applying the hash function to *Alice*'s identifying data, he obtains *Alice*'s anonymous identifier: AID_{Alice}. Then, *Bob* compares AID_{Alice} with the anonymous codes of the *AMDB*.

In a dictionary attack, Bob applies the hash function to a list of identifying data (a dictionary) and draws-up a table linking identifying data with the corresponding anonymous code. A simple comparison with the *AMDB* could lead to deduce the medical data related to the persons of the dictionary. Concerning the linkability property, it is obvious that even after anonymization, it is always possible to identify if two information items belong to the same person. Unlinkability is thus not satisfied.

In order to avoid such vulnerabilities, two keys have been added before applying the hash function. The first pad, k_1, is used by all senders of information as follow "$Code_1 = \mathbf{H}(k_1 \mid Identity)$"; and the second, k_1, is applied by the recipient "$Code_2 = \mathbf{H}(k_2 + Code_1)$". Nominal information is therefore hashed twice, consecutively with these two keys. The aim of pad k_1 (*resp.* k_2) is to prevent attacks by a recipient (*resp.* a sender) (Figure 18.2).
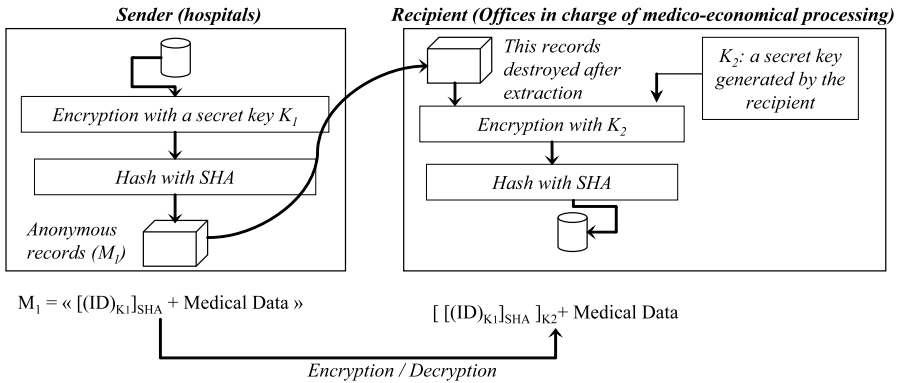
Fig. 18.2 Anonymization in French hospitals

However, this protocol is both complex and risky: the secret key should be the same for all information issuers (clinicians, hospitals, etc.) and stay the same over time. Moreover, this key must always remain secret: if this key is corrupted, the security level is considerably reduced. It is very difficult to keep a key that is widely distributed secret during a long time. This means that new keys have to be distributed periodically. The same applies when the hash algorithm (or the key length) is proven not sufficiently robust any more. But, how can we link all the information concerning the same patient before and after changing the algorithm or the key? If this problem occurs, the only possible solution consists in applying another cryptographic transformation to the entire database, which may be very costly.

Besides, note that at the recipient side (e.g., for medico-economical or statistical studies), it is sometime not desirable to link data belonging to the same patient. However, even after adding the two pads, this solution does not ensure the unlinkability property.

18.2.2.3   Example of Anonymization in Switzerland

The Swiss Federal Office for Statistics (SFSO) is responsible for collecting medical data on all individuals hospitalized in Switzerland. Information on the diagnoses and on the corresponding treatments is given for all patients.

The first solution proposed by the SFSO was to slightly hide the identifying data (for example, the name of the patient was replaced by its SOUNDEX code). However, the Swiss Medical Computer Science Society (SSIM) reacted very negatively to this first project. They argued that this new statistic would create a large database that would not preserve the privacy of the patients' medical records.

The SFSO therefore contacted the Swiss Federal Section of Cryptology (SFSC) to find solutions to this problem [JJC01]. A primary analysis of statistical needs conclude that it is not necessary to know to whom a given record belongs; but the SFSO needs to recognize that two different records actually belong to the same person (the linkability property). This is crucial in order to follow the history of the patients. A second analysis leads to split data into two categories: medical data (diagnosis, treatment, . . .) and non-medical data (last name, first name, date of birth, domicile, . . .).

In order to preserve the anonymity, the level of precision of non-medical data has been reduced to the minimum needed for the statistics; for example, they use the age instead of the date of birth, or the region instead of the domicile.

The non-medical data that are really identifying (personal data) are not used directly in the statistics. Essentially, these identifying data are replaced by a calculated personal code, called anonymous linking code, which characterizes the patient without revealing his/her identity; it satisfies the following properties:

- the same person always receives the same personal code,
- the identifying data allow one to calculate easily the personal code of a patient,
- two different people always receive two different personal codes (no collision),
- the personal code of a patient does not allow his/her identification (robustness).

The next step was to choose on which identifying data the calculations will be based. On the one hand, these data should always be available and should stay constant over time; on the other hand, collisions should be avoided. Finally, the identifying data was restricted to: date of birth, gender, last name and first name. These identifying data is replaced by a fingerprint, called anonymous linking code: $fingerprint = \mathbf{H}\,(ID\text{-}Data)$, with H being a secure hash function (Figure 18.3).

Before transmitting medical data to the SFSO, the hospital generates a session key c; this key is then used to encrypt the fingerprint during the transmission: $\mathbf{IDEA}(fingerprint)_c$; a public key cryptosystem is used to transmit the session key $\mathbf{RSA}(c)_E$ using the SFSO public key E.

After reception, c is retrieved by using the SFSO private key D; the encrypted fingerprints are then decrypted, and uniformly re-encrypted by the symmetric key K of the SFSO: they become the anonymous linking codes used as personal codes. The key K is distributed among several trusted persons, using Shamir's secret sharing technique [Sha79].

However, it is easy to notice that the intermediate steps of these transformations should never be visible to the SFSO operators. Indeed, how can we be
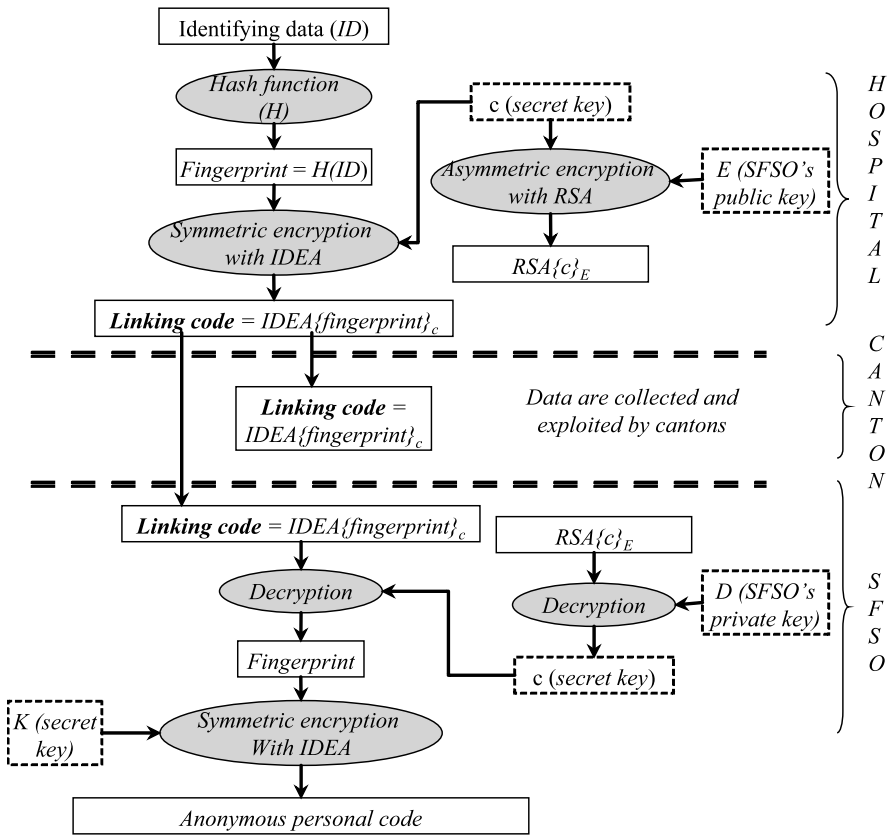
Fig. 18.3 Transformation of identifying data in Switzerland

sure that the secret key c and the fingerprints are never recorded in a storage medium? It is clear that these steps (calculation phases) should be done in a well-protected hardware module (a kind of secure "Black-box"). In addition, inviolable access control mechanisms (e.g., specific tamperproof hardware), could improve the protection. The aim is that only trustworthy persons, acting together, should carry out the composite operation. Besides, as for the French solution, this procedure is only specific to uses where unlinkability is not needed, while in medical fields this property is sometimes important.

18.2.2.4 Example of Anonymization in Germany

The German National Cancer Registry was founded in 1953, with the aim of gathering medical statistics related to German cancer cases. Nowadays, it includes detailed information on 2 million cancer patients. The following sensitive personal and medical information were recorded for each cancer case:

- cancer patient's personal identification;
- tumour location, histology, stage, diagnosis, and therapy;
- further treatment and follow up;
- individual and family history;
- death, including autopsy results, if any.

The German anonymization procedure is a little bit similar to the SFSO's solution (even if the purposes of uses are different).

Actually, the procedure of the population-based cancer registration is realized in two steps by two institutions [Blo97]. In the first stage, the Trusted Site accumulates the patient-related tumor data recorded by doctors, dentists or Follow-up Organization Centers. The Trusted Site anonymizes the cancer patient's personal data by an asymmetric procedure, e.g., a hybrid IDEA-RSA encoding (like in the Switzerland solution). The identifying data is encrypted with an IDEA session key, generated randomly. The IDEA key is encoded by a public RSA key. To allow an unambiguous assignment of additional information to the correct patient record, a control number (a kind of pseudonym) is generated, using different attributes of the personal data that are sufficient to identify uniquely each patient. This control number is generated by using a one-way hash function (MD5) and a symmetrical cryptography algorithm (IDEA). To allow the assignment of data from the different federal Bundesländer, the control number procedure and key are unique for all of Germany ("Linkage Format"). The Trusted Site transfers both the encrypted patient-identifying data and the epidemiological plaintext data to the Registry Site.

The latter stores the record in the register database and brings together different records belonging to one patient. After the matching of data, a random number is added to the control number and the result is symmetrically encrypted by IDEA ("Storage Format"). To match new records, the control numbers must be transformed back from the "Storage Format" to the "Linkage Format".

As the security mechanisms used in this solution are quite similar to those used in the Switzerland procedure, these two solutions have the same limits.

Table 1 summarizes the characteristics of the examples explained above.

Table 18.1 Summary of existing solutions

Country	Purpose	Technique	Property
USA	Social security data processing	Secret keys (Tricryption)	Anonymity
France	Linking medical data for evaluation purposes	Symmetric keys + hashing	Anonymity + Linkability
Germany	Statistics	Hybrid encryption + hashing	
Switzerland			

18.2.3   Requirements for a Suitable Implementation

18.2.3.1   Global Needs

Most of the existing solutions have been developed empirically, concern only one specific use and generally enforce one privacy property. However, fine-grained privacy often requires complex mechanisms related to several needs (anonymity, pseudonymity, unlinkability, etc.).

Let us take some examples from the French regulation.

In order to evaluate regional as well as national healthcare activities, the doctors send medical as well as personal data to an authority called the "professional unions". These data are anonymous (anonymity of the patients' and the doctors' identities). However, in specific situations, this authority needs to evaluate the physician's behavior and to assess care quality; it should thus be possible to re-identify the concerned physician. In these cases, it is the matter with pseudonymization. Let us take another example; some diseases have to be monitored, through statutory notification, to evaluate the public health-care policy (e.g., AIDS) or to trigger an urgent local action (e.g., cholera, rabies). The treatment of these data (e.g., for prevention, care providing, epidemiological analysis) requires both anonymity and linkability.

We can give many more examples, but the general conclusion are: as current systems (and laws) grow and change more often, there is a real need to a systematic methodology. The latter should be generic, evolvable and easily adapted (and parameterized) to satisfy the requirements of particular situations.

18.2.3.2   A Systematic Methodology

Traditionally, the security analysis process studies two main phases:

- the *request* (demand) in the form of needs to be satisfied;
- the *response* in the form of functionalities and solutions.

Basically, before deriving an anonymization solution, we have suggested three complementary levels of analysis [KD05, KD06]:

- The *anonymization needs* represent the user expectations; generally, their form is neither very explicit nor very simple to formalize.
- The *anonymization objectives* specify the security level to reach, the information to protect, the threats (against privacy) to counter, etc.
- The *anonymization requirements* represent how to express the needs (and the threats to counter) with a non-ambiguous semantics and to characterize the solutions to implement.

Anonymization Needs

The *anonymization needs* represent the user's expectations to obfuscate any information that could be exploited to disclose his/her private information. Depending on the system, the context, the purpose, the environment, etc., some information can be considered private or not, and sensitive or not. Generally, directly as well as indirectly, nominative data should be anonymized; in some cases, even a date of birth or a zip code can be considered as sensitive.

If we take again the example of AIDS data, the study of the anonymization needs should clarify if the aim is to globally evaluate the impact of prevention activities; or to finely evaluate the impact of therapeutic actions as well as a follow-up of all the cases.

This kind of analysis has important consequences on the nature of data to be collected and/or anonymized. Currently, as the aim is a global evaluation of AIDS cases as well as an epidemiological surveillance of the HIV evolution (at the regional level), the need's analysis leads to the following conclusions (related to data impoverishment):

- Instead of collecting the zip code, it is more judicious to collect a region code.
- Instead of collecting the profession, we think that a simple mention of the socio-professional category is sufficient.
- Instead of mentioning the country of origin it is sufficient to know if the HIV positive person has originated from a country where the heterosexual transmission is predominant, etc.

Finally, note that in most cases, the privacy-related regulations (cited in the introduction) could serve as a good input for this step by providing global privacy needs (that should be refined and adapted to particular situations). For instance, the resolution A/RES/45/95 of the General assembly of United Nations defines (as well as the recommendations of the Commission on Human Rights resolution 1990/42 of 6 March 1990 and Economic and Social Council resolution 1990/38 of 25 May 1990, entitled *Guidelines on the use of computerized personal files*) defines the following needs (in the form of guidelines and principles) related to privacy:

- *The principle of accuracy*: the persons responsible for the compilation of files or those responsible for keeping them have an obligation to conduct regular checks on the accuracy and relevance of the data recorded and to ensure that they are kept as complete as possible.
- *The principle of the purpose specification*: the purpose which a file is to serve and its utilization in terms of that purpose should be specified and legitimate, and, when its utilization is established, it should receive a certain amount of publicity or be brought to the attention of the person concerned, in order to make it possible subsequently to ensure that:

  – All the personal data collected and recorded remain relevant and adequate to the purposes so specified;
  – None of personal data is used or disclosed, except with the consent of the person concerned, for purposes incompatible with those specified;
  – The period for which the personal data are kept does not exceed that which would enable the achievement of the purpose so specified.

- *The principle of interested-person-access*: everyone who offers proof of identity has the right to know whether information concerning him is being processed and to obtain it in an intelligible form, without undue delay or expense, and to have appropriate rectifications or erasures made in the case of unlawful, unnecessary or inaccurate entries and, when it is being communicated, addressees.
- *The principle of non-discrimination*: arbitrary discrimination, including information on racial or ethnic origin, colour, sex life, political opinions, religious, philosophical and other beliefs as well as membership of an association or trade union, should not be compiled.
- *The principle of security*: appropriate measures should be taken to protect the files against both natural dangers, such as accidental loss or destruction, and human dangers, such as unauthorized access, fraudulent misuse of data or contamination by computer viruses.
- *Supervision and sanctions*: the law of every country shall designate the authority which, in accordance with its domestic legal system, is to be responsible for supervising observance of the principles set above.
- *Transborder data flows*: when the legislation of two or more countries concerned by a transborder data flow offers comparable safeguards for the protection of privacy, information should be able to circulate as freely as inside each of the territories concerned. If there are no reciprocal safeguards, limitations on such circulation may not be imposed unduly and only in so far as the protection of privacy demands.

Anonymization Objectives

Once the needs (request) are well-identified, we can now start studying the solution characterization (response). In this respect, the first step is to identify the anonymization objectives. Basically, we should answer the question: "What kind of anonymization to use?" We thus define the *anonymization objectives* according to one of the three following properties, applied to the anonymization function:

- *Reversibility*: hiding data by encryption. In this case, from encrypted data, it is always possible for legitimate authorities, by using some secret key, to retrieve the corresponding original nominative data (decryption), and conversely (encryption).

- *Irreversibility*: the property of real anonymization. The typical example is a one-way hash function: once replaced by anonymous codes, the original nominative data are no longer recoverable.
- *Invertibility*: this is the case where it is, in practice, impossible to re-identify the person, except by applying an exceptional procedure restricted to duly authorized users. This exceptional procedure must be done under surveillance of a highly trusted authority like the medical examiner, the inspector-doctor or a trustworthy advisory committee. This authority can be seen as the privacy guarantor. Actually, it is a matter of a pseudonymisation according to the common criteria terminology [ISO06].

Anonymization Requirements

Defining if the anonymization function should be invertible, reversible or ir-reversible is not sufficient. Indeed, sometimes, even after anonymization, at-tacks by inference (or by dictionary) are able to re-identify the person. In this respect, two kinds of requirements must be satisfied by the anonymization system: the "linking" requirements and the "robustness" requirements.

Linking allows for associating (in time and in space) one or several anony-mous codes to the same person. As mentioned in Figure 18.4, the linkage can be temporal (always, sometimes, never) or geographic (international, national, regional or local). We should thus identify—at this step—which data should be linked, by which entities (users, organizations, systems etc.), for which time and at which level.
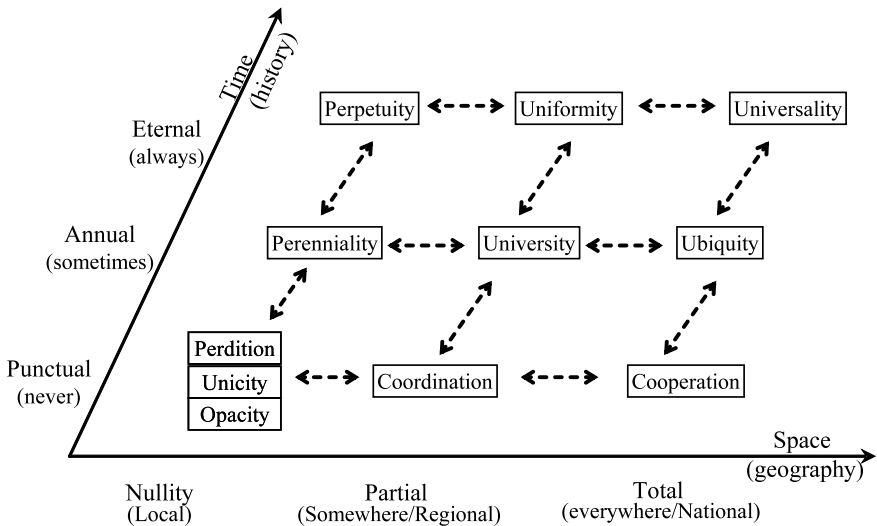
Fig. 18.4 Network of the anonymization cases

The robustness requirements, concerning exclusively illicit deanonymization, can be divided into two distinct cases: robustness to reversion and robustness to inference.

- The *reversion robustness* concerns the possibility to inverse the anonymization function, for example if the used cryptographic techniques are not strong enough. For instance, in the German example cited in Section 18.2.2.4, healthcare providers use 640 bit RSA keys. Nowadays, this cryptographic function does not satisfy the reversion robustness requirement.
- The *inference robustness* concerns data deanonymization by means of unauthorized nominative calculation. We identify several kinds of inferences:
    - `deductive`: it consists of inferring, mainly by first-order logic calculation, unauthorized information on the only basis of publicly-available data; for example, if a particular patient does a genetic screening test, and few days later he makes a dosage test, then we can deduct that the screening test was positive.
    - `inductive`: when the conventional reasoning that uses information explicitly stored in the information system is not sufficient to infer information, this reasoning can be completed by making some hypothesis on certain information;
    - `probabilistic`: it consists of inferring, by stating a set of various plausible assumptions, an unexpected secret information from valid available data. For example, as the patient P is treated in a particular hospital specialised in diseases *M1* and *M2*, and as the probability to have *M1* is very small (10 %) in its age; then we can presumably deduct that p is suffering from *M2*.

This list is not exhaustive, and naturally, we can imagine other types of inference channels based on other types of reasoning.

Solution Characterization

For a certain scenario, once the privacy needs, objectives and requirement are defined, one can characterize the most suitable solutions (that responds to the identified needs and that satisfies the identified objectives and requirements). More precisely, the following needs to be considered:

- The *type of the solution* to develop: Is it an organizational procedure, a cryptographic algorithm, a one-way function, or a combination of subsets of these solutions?
- The *plurality of the solution* to implement: Do we need one anonymization system, double or multi-anonymization procedures (e.g., by taking into account the linking and reversion requirements)? Of course, the choice is related to the type of anonymization function inversion threats (direct or indirect reversion).

- The *interoperability of the solutions* that are to be combined: transcoding (manually, for some continuity reasons) or translating (mathematically, for some consistency reasons) an anonymization system of anonymous identifiers into another anonymization system; or transposing (automatically) several anonymization systems into a unique anonymization system of anonymous identifiers, in order to authorize or forbid the matching of anonymized data.

Figure 18.5 illustrates our methodology by presenting the anonymization taxonomy.
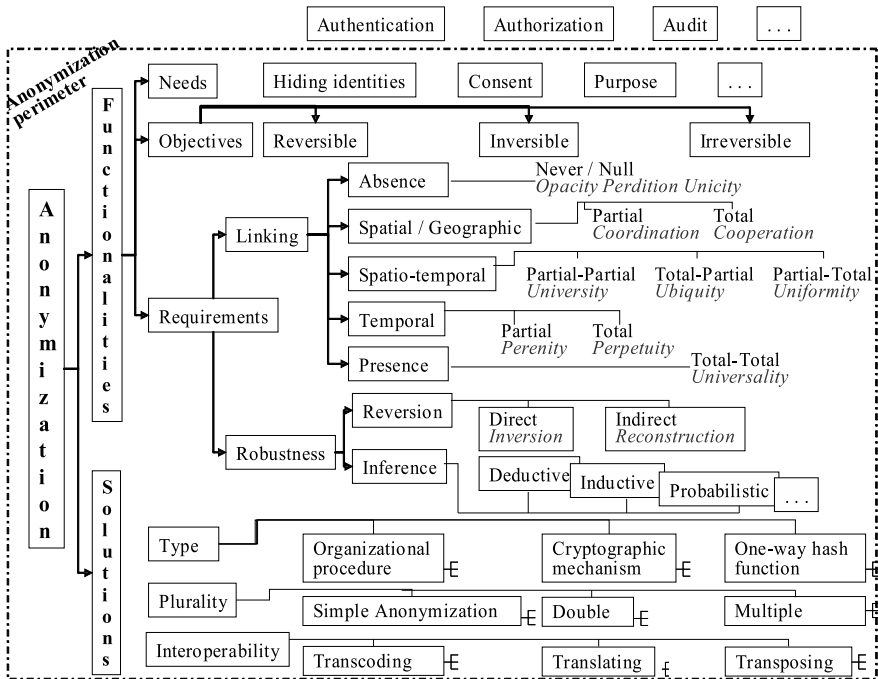
Fig. 18.5 The anonymization taxonomy

In this section we have defined a systematic approach for data anonymizations. In the next section we suggest a generic architecture that satisfies the raised requirements and summarizes the possible use cases.

18.2.4   A Generic Anonymization Architecture

In order to emphasize the different notions presented below, we use a three-level architecture. Note that even if we explain it through a healthcare

example, the global logic can be followed for other fields. Actually, our architecture distinguishes three kinds of organizations (Figure 18.6):

- Hospitals, clinics and healthcare providers: organizations and users that collect medical and personal data, essentially for epidemiological uses;
- Processing centers: organizations that essentially use medical data for medico-economical studies such as, statistical administrations, research organizations, etc.;
- General and public users or organizations such as press, research labs, web publishers.

According to the privacy-related regulation R(97)5 [R(997], the anonymization needs study leads to distinguish three kinds of databases in hospitals: administrative, medical and one or several anonymized databases. Each anonymized database contains the information for a particular project. A project is a program or a study intended for statistical, epidemiological, therapeutic research or medico-economical data processing.

At this step we can identify the following needs, objectives and requirements:

- Needs: three databases; patient consent when using his medical data, and so on;
- Objectives: invertibility, e.g., when the end user (e.g., researcher in rare diseases) discovers important information that requires re-identifying the patients;
- Requirements: robustness to reversion and to inferences; linkability in the projects / processing centers of data belonging to the same patient (i.e., each project can link data corresponding to the same patient, even if they come from different hospitals); unlinkability and unobservability of data processed in different projects / processing centers. Other requirements will be derived during our analysis.

To achieve these requirements, we suggest the generic architecture of the Figure 18.6.

In this architecture, the transition from a medical database to an anonymized one requires the application of two transformations (**T1**, **T2**).

T1: consists of calculating "IDApat | Proj", an anonymous identifier per person and per project. "IDproj" is the project identifier; while "IDpat" is the permanent patient anonymous identifier (a random number). To satisfy the regulation cited above (e.g., [R(997]), we suggest that IDpat is held under the patient's control, e.g., on his personal medical smart card.

In the hospital, when transferring data to anonymous databases, the user (i.e., the healthcare professional) sends IDproj to the card. The card already contains IDpat. By supplying his card, the patient gives his consent for his data to be exploited as part of this project. Fore more security, we suggest that the T1 procedure, run within the smart card (tamper-resistant), consists
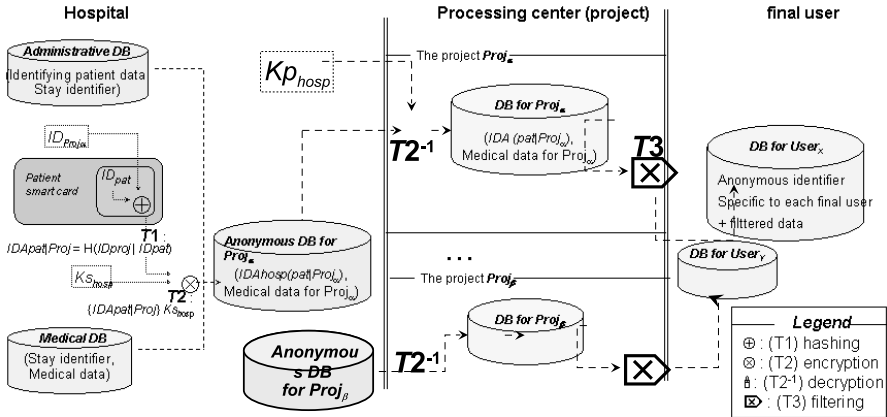
Fig. 18.6 The suggested anonymization procedure

in applying a one-way hash function (e.g., SHA-2) to the concatenated set (IDproj | IDpat):

$$(\mathbf{T1})\ \text{IDApat|Proj} = \mathbf{H}(\text{IDproj | IDpat})$$

Nevertheless, the transformation **T1** does not protect against attacks where attackers try to link data held by two different hospitals. To make this clearer, let us take an example where *Paul* has been treated in the hospitals *HospA* and *HospB*. In each of these two hospitals, *Paul* has consented to give his data to the project $Proj_p$. Let us assume that *Bob*, an *HospB* employee, knows that the fingerprint X (= IDAPaul | $Proj_p$) corresponds to *Paul*, and that *Bob* obtains (illicitly) access to the anonymous database held by *HospA* and concerning $Proj_p$. In this case, the malicious user *Bob* can easily establish the link between *Paul* and his medical data (concerning $Proj_p$) held by *HospA* and *HospB*.

In order to face this type of attacks, a cryptographic asymmetric transformation (**T2**) is added. Thus, before setting up the anonymous databases (specific to each project), the hospital encrypts (using an asymmetric cipher) the fingerprint IDApat|Proj with the encryption key Kshosp specific to the hospital; (the notation "$M\mathbf{K}$" indicates that M is encrypted with key K):

$$(\mathbf{T2})\ \text{IDAhosp(pat|Proj)} = \{\text{IDApat | Proj}\}\mathbf{Kshosp}$$

If we take again the previous scenario, the malicious user Bob cannot re-identify the patients because he does not know the decryption key KphospA.

Kshosp and Kphosp are a key pair of a public key cryptosystem, but that does not mean that Kphosp is really public: it is known only by the project processing centers and by the hospital's security officer (who knows also Kshosp, of course).

Basically, the anonymous databases intended to one or several projects are periodically sent by hospitals to processing centers. Let us recall that a processing center could be an association, an office for medical statistics, or a research center.

When anonymized data are received from a hospital by a processing center, these data undergo transformations that depend on IDAproj|pat and on Kshosp. Every center decrypts received data by using Kphosp:

[IDAhosp(pat|Proj)]**Kphosp** = [{IDApat|Proj}**Kshosp**]**Kphosp** = IDApat|Proj

Note that since the resulting data are associated to IDApat|Proj, each project can link data corresponding to the same patient, even if they come from different hospitals.

Before their distribution to the final users (statistics organizations, web publishing, press, etc.) the anonymized data can undergo a targeted obfuscation. This obfuscation can consist in filtering out data that are unnecessary for the intended processing, or in reducing the accuracy of some useful data, e.g., replacing a birth date by an age range or a zip code by a less precise area code. This should be achieved through some trade-off: the reduction of accuracy should be sufficient to prevent malicious inference, but the data should stay accurate enough to enable the authorized processing.

$$(\textbf{T3})\ \text{IDApat|util} = \textbf{H}\ (\text{IDApat|Proj} \mid \text{Kutil|proj})$$

In accordance with the needs, this transformation can provide different linkability properties:

- if the aim is to allow full-time linking, Kutil|proj has to be stored by the processing center and reused for each distribution to the same user;
- conversely, if the center wishes to forbid the user to link data distributed by the center at different times, the key is randomly generated just before each distribution.

18.2.5   Implementation

Figure 18.7 describes a very-simplified architecture of our implementation. Our prototype has been tested on several platforms: Unix (CPU: HP 9000/ L2000; OS: HPUX 11.00) Linux (CPU: Athlon 1.4 Ghz; OS: Mandrake Linux 10; kernel v2.6.3.4) Macintosh (CPU: Power G3 600MHz; OS: MacOS X 10.x) and Windows. The prototype requires about 50 Mbytes free disk space, MySQL or Oracle 10g, Java JRE 1.4.2 or later and JavaCard 2.1. SUN recommends a configuration with 1 KB of RAM, 16 KB of EEPROM, and 24 KB of ROM [KD05] [Che00].

For the implementation, it is advisable to use cards that support cryptographic procedures (smartcards). The kit should provide a complete environment of development and should contain an interface that makes the

communication with the smart card easier (e.g., the "JCardManager" inter-
face provided in the Gemexpresso RAD III kit). Programming is done in the
standardized language "Javacard". Javacard is a reduced-API Java. Even if
this API provides most characteristics of Java (e.g., exceptions, constructors,
inheritance, unidimensional arrays), it has some limits, insofar as the primitive
types are limited to byte, shorts, and boolean; it does not support cloning,
threads, garbage collector, etc.

In order to use the application, you should have some software components
such as: the patient cards, the Healthcare Professional certificates (embedded
in the "Healthcare Professional" cards), the hospital keys (public and private
keys) and the certification authority public key.
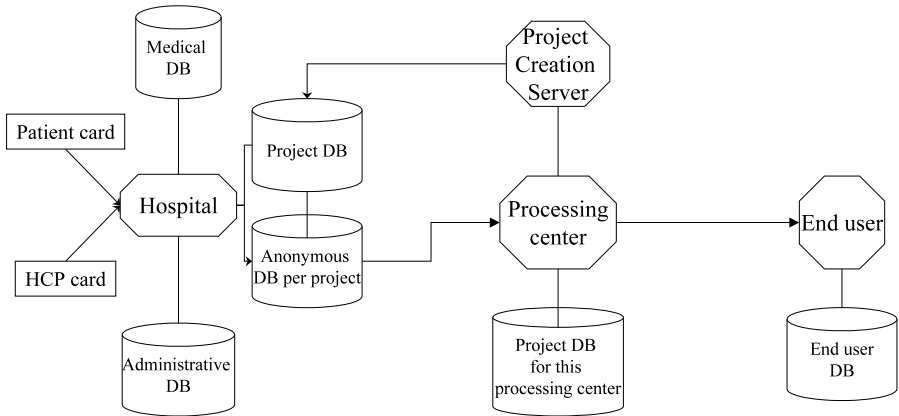
Fig. 18.7 Architecture scheme

18.2.6   Discussion

The solution that we propose guarantees several benefits. First, it is fine-
grained, generic enough and could be easily adapted to different sector needs
(e.g., social domain, demographic projects, etc.).

Second, the use of smartcards (that are sufficiently tamper resistant) helps
to keep the sensitive data (e.g., the patient identifier IDpat) secret and to pro-
tect the critical processes. Moreover, the secret as well as the anonymization is
held under the patient's control. The patient's consent is required for each gen-
eration of an anonymized form of his personal data. Indeed, the medical data
can appear in a certain database only if, by supplying his card, the patient
allows the use of his medical data as part of a certain project. The patient's
consent is also necessary when reversing the anonymity. Let us take the ex-
ample when the end user (e.g., researcher of rare diseases) discovers important

information that requires re-identifying the patients. At first, it sends back results to the hospitals participating in the project (e.g., a given orphan disease study). When the patient produces his medical data card (which implies that he gives explicitly his consent), it is possible to calculate IDApat|Proj = $\mathbf{H}$(IDproj | IDpat) and IDAhosp(pat|Proj) = {IDApat|Proj}$\mathbf{Kshosp}$, and to establish the link between the patient, his anonymous identifiers, and his medical data. A simple (and automatic) comparison between the anonymous identifier and the inversion list would trigger an alarm. This alarm asks the patient if he wants to consult the results.

Third, the solution resists dictionary attacks that could be run in various organizations: hospitals, processing centers or end users.

Fourth, contrary to existing solutions (e.g., the current French anonymization procedure), in our solution, the identifiers (IDproj, IDpat, IDApat|Proj and IDApat|util) used in the various transformations are located in different places. Similarly, the keys (Kshosp, Kphosp) are held by different persons. Indeed, IDproj concerns a unique project; the pair (Kshosp, Kphosp) is specific to one hospital; IDApat|util is dedicated to a single end user. Moreover, IDpat is specific to one patient, and only held on his card. Thus, even if a certain IDpat (corresponding to Paul, for example) is disclosed (which is not easy!), only Paul's privacy could be endangered (but not all the patients' privacy, as it is the case in the current French procedure) and only for certain projects. Finally, it is possible to merge data belonging to several establishments without compromising either security or flexibility. Indeed, if two hospitals (HospA and HospB) decide to merge someday, it would be easy to link data concerning every patient that has been treated in these hospitals. In fact, each hospital would have to decrypt its data with its key Kphosp, and then encrypts the result by KshospAB the new hospital private key. Furthermore, according to the security needs of the studied cases, we suggest complementing our solution by other technical and organizational procedures: In particular, the access to data has to be strictly controlled; a well-defined security policy must be implemented by appropriate security mechanisms (hardware and/or software). Reference [KBM$^+$03] suggests a security policy and an access control model (Or-BAC: Organization-Based Access Control) that are suitable to collaborative systems. Indeed, Or-BAC offers the possibility to handle several security policies associated with different organizations.

It is also recommended to control the purpose of use by implementing intrusion detection mechanisms. In particular, these mechanisms should easily detect sequences of malicious requests (illicit inferences, abuse of power).

18.2.7 Conclusions

Data anonymization is a critical issue in many emerging applications and networked systems, in particular in the healthcare domain. International regulation authorities as well as computer science communities are worried by

this problem. This chapter presents an analytic approach and a generic solution to protect personal and sensitive data through anonymization. It also gives details of a possible anonymization architecture and its implementation. Further work would be needed to assess as precisely as possible the sensitivity of anonymized data, i.e., to estimate how easy it would be to infer sensitive personal data from anonymized data. Such studies have already been applied, e.g., in statistical databases. They can be based on information theory, but they should also take into account precise characteristics of the use of the anonymized data. For instance, in a statistical database, the attacker can adapt his/her queries to target specific data; this is not possible with anonymized data.

18.3   Anonymous Communication

Confidentiality and data protection in communication networks has become more and more important, ever since the development of the Internet from the ARPA-net and its growing number of users. In contrast to cryptography, which deals with content protection, anonymity is about hiding relationships between communicating peers. To this end, anonymization networks are designed and developed in order to achieve anonymity for senders, recipients, or both at the same time. Here, the term *anonymity* is defined as "the state of not being identifiable within a set of subjects, the anonymity set" [PH06], i.e. the users are protected by means of a set of other users in order to avoid their identification.

Providing an anonymous communication service for the Internet is a demanding task. While cryptography can be used to protect integrity and confidentiality of the data part of the packets, everyone along a route of packets can observe the addresses of the communication partners. To achieve anonymity against third parties, a packet's source and destination addresses must be hidden and its appearance should vary from hop to hop. Moreover, timing correlations should be thwarted in order to provide protection against attackers that are able to observe large portions of networks and therewith have a good overview of the traffic within.

There is a number of proposals and practical implementations of anonymization networks (see e.g. [Ray00]). Most of them are based on mixing [Cha81], onion routing [DMS04], or on DC-networks [Cha88]. A number of attacks exist, especially on the low-latency implementations (c.f. [MD05b]) that are not trivial to defend against. Those based on the DC-networks can be used to provide perfect anonymity under some assumptions [Cha88], however the protocol has serious drawbacks causing questionable practical implementation, i.e. it requires secure and reliable broadcast channels, is prone to channel jamming, inefficient in large networks, etc. [Ray00].

Anonymous communication is a basic fundamental building block for privacy-friendly web browsing, any viable identity management system, privacy-aware

eGovernment, eCommerce and eHealth technologies, as well as for providing freedom of speech, mind, and achievement of democratic principles even in those countries that try to filter and censor access to information.

18.3.1 Scenario

This section is about describing the scenario and the setting of anonymous communication. Its purpose is to explain which items, entities, and properties are important in this context, and why. To a certain extent, this section will also try to explain why certain issues are usually not regarded as important in the core field of anonymous communication.

As data streams in the Internet are usually initiated by people and directed towards people, we will assume in the remainder of this text that there is a person called "Alice" which wishes to communicate to some "Bob". As long as there is nobody else around, their communication is safe and sound.

However, if some "Eve" arrives and wishes to determine, to whom Alice is talking, Alice will have to take precautionary measures, i.e. start to use tools for anonymous communication in order to hide her communication relationship to Bob.

Now, Alice and Bob communicate over middle-men, or with other means, such that there is no direct link to be established between them.

From this small example set, we can already derive next to all important questions in this research area:

- Who are Alice and Bob? Why do they hide their communication? What powers do they have?
- Who is Eve? How powerful is she?
- Who are the other people involved, the middle-men?
- What does to general public think of all of this?

We will answers these questions in the upcoming section in the given order.

18.3.1.1 Users of Anonymizing Networks

This section tries to answer what kind of people are using anonymizing networks, and analyse their motivation to do so. As users of an anonymizing network, nearly by definition, are next to impossible to interview, one of the few scientific methods of getting hold of their interests, is to read the output of such a network. This can then be used in order to create and deduce theories about the people creating this kind of traffic.

Several prior analyses of Tor network traffic exist. There have been both user surveys adressing the users of anonymization services on the web [Spi03] and observatory approaches, relying on the classification of logged traffic into several categories [Fed05]. However, the results of the two types of study seem to be somewhat contradictory, concerning both background of usage

(with self-reports overstating professional use compared with the measurement/categorization approach) and use cases. While this discrepancy may be explained with the well-documented bias of people to overstate their privacy sensitivity (for an overview see [Acq04, Syv03]), or the generally weak validity of self-report studies in the context of sexuality [OFB97], to our knowledge there is currently no better public study.

However, the material available allows the conclusion that most users of anonymizing networks are either

- seeking political Information, which is illegal or hard to obtain in their countries of origin,
- surfing or gathering pornographic material, an action which is severely punished in some countries of this world, and at least socially unacceptable in most others,
- attacking other computers in the Internet,
- using the networks for their everyday business on the Internet.

Please note, however, that the list as presented above does neither claim to list all purposes, nor is the proportion of people doing an action correlated to the position as given.

In general it is true that due to the currently missing quality-of-service in anonymizing networks, the majority of users are really in need of hiding their communication patterns.

18.3.1.2   Attackers on Anonymizing Networks

As anonymizing networks are a protective measure, it is always important to take into account against which adversary a user, or a group of users, wants to protect itself.

There are a number of attacker models in the traditional literature of anonymous communication. Most of them are either very simplified or pretty abstract – therefore difficult to generalize or even identify in real networks [PP06a]. Often such a model is abstract, unsystematic and it is not trivial to identify the exact threats for the end-user of the implemented system. In the following we introduce a classification of attacker types for the risk analysis and attacker modelling in anonymous communication independently of the concrete technique. The classes are designed in such a way that their meaning can be easily communicated to the end-users and management level. We claim that the use of this classification can lead to a more solid understanding of security provided by anonymizing networks, and therewith improve their development.

Especially in the field of anonymous communication there exist a large number of attacker models. Most of these are describing the actual capabilities of the attacker, not considering the capabilities needed in real life to achieve the proposed capabilities. A common example is the passive global

observer. We agree that this model is needed and interesting for mathematical analysis, however users should be aware that theoretical results based on this analysis are not representative in real scenarios: an attacker having the capabilities to intercept traffic at the global scale can typically also easily alter and manipulate the traffic and, therewith invalidate the results of the analysis and protection vision of the end-user. From another perspective, it is not realistic for an average end-user to defend against an adversary that is capable of observing the whole worldwide network, because such a powerful adversary can make use of more efficient means in order to obtain the same information.

The attributes that distinguish most real-life attackers are the *amount of computational power* and the *amount of influence that the attacker has on the network*. The latter correlates most often with the number of nodes and links that the attacker controls or which are within his reach. Furthermore, computational capabilities are not as relevant in today's scenarios because cryptography is usually too strong to be broken by anybody but governmental agencies and computational breaking of other mixing is only seldom preliminary to attack an anonymizing system.

It is assumed as an unconditional requirement that the user's terminal is under his own control and cannot be compromised by any other party. Otherwise it is trivial breaking the user's privacy and anonymity.

0. External Party: The least powerful attacker has no control of any computer and no network link between the two communicating parties. While this kind of attackers are hardly worth being called so, there should be still taken measures to prevent them from gaining information.

Note that external parties can be very powerful, e.g. competitors in international trade, or organized crime. But unless further actions are taken to increase their influence on anonymizing networks, their influence is limited.

1. Service Provider: This class of attacker represents the user's communication partner. In some scenarios it is desirable to use service without disclosing the sender's true identity, thus in these cases, the receiver of the message can be considered a potential attacker. This attacker is technically bound to the receiving end of the communication and its close neighborhood.

2. Local administration: This class of attackers can manipulate and read everything in the close network environment of the user.[7] These capabilities can be very powerful if the user blindly trusts all the transmitted and received data or does not care about protection. On the other hand, this attacker can be easily circumvented once the user is able to establish a secure connection to an outside trusted relay.

---

[7] Think of sniffing data, manipulated DNS-responses, man-in-the-middle attacks on TLS-secured connections, denial of access to anonymizing networks to force plain communication, and much more.

3. ISP: The next powerful attacker has access to the significant larger area of computers in the vicinity of the user. The amount may be so large that it can even be a non-negligible part of the whole global network. It is thus possible that a major number of relays on the way to the communication partner is within the reach of this class of attacker.

4. Government: This adversary has the power to access not only a significant portion of all networks but also has large resources to fake services, break simpler encryption schemes[8] or prohibit access to specific services. This adversary might also take measures that violate existing laws to a certain extent and has the power to draw significant advantages from doing so.

5. Secret Services: are forming the highest class of an adversary. They can be assumed to either have access to most parts of the global networks or they can get the access if they think it is necessary for their operation. This class of attacker is also not bounded by any kind of laws. It should be mentioned that the latter two types of attackers will probably not refrain from using non-technical methods to get information – this includes but is not limited to the physical capture of nodes and people. It is noteworthy that some countries deploy their Secret Services for industrial espionage.

We deliberately don't specify the classes of attackers in more detail, but rather leave them as categories that are intuitively understood by researchers as well as by the end-users. Note that these classes must not be strict: seamless transition is possible.

From our point of view, the minimum requirement for an anonymizing network should be to defeat from attackers of class 0 upwards to the class 2 or 3. While it seems currently to be infeasible and to some people not desirable to protect all end-users from attackers of class 4 and higher ones, we list these for completeness reasons and because there exist users that want to defend themselves from this kind of adversaries.

18.3.1.3   Operators of Anonymizing Networks

Currently there are only two major groups of anonymizing networks in operation: those where the nodes are operated by privacy enthusiasts and those which require the user to register and pay for the service.

The former covers the networks of Tor, I2P, freenet, and Mixmaster, while the later set contains the networks of Jondos/AN.ON, and commercial single hop proxy providers. Especially for the operators of the second set of networks, the motivation to provide the service is clearly the financial gain.

On the contrary, there is no know research that investigates why and to which extend operators run and deploy nodes for networks. An answer to this question would however be especially interesting, as they do not only

---

[8] The German Federal Office for Information Security factored the RSA-640 number in September 2005 and single-DES is known to be weak for decades: `http://www.rsasecurity.com/rsalabs/node.asp?id=2092`

offer computing power and bandwidth to other users, but are sometimes also taken into legal responsibility for the actions which are relayed through their computers. We will continue discussion on this topic in Section 18.3.4.

18.3.1.4   Third Parties in Anonymizing Networks

In addition to the entities directly involved in the communication, i.e. sending or relaying messages, or being a known peer partner of Alice, there are sometimes other entities included in the scenario.

This can be, for example, an ISP, which objects to the traffic usage generated by nodes of anonymizing networks, which can in some setting easily grow up to several Terabytes per month.

Another entity involved can be the owner of copyrighted material, which is exchanged in an illegal manner through such means of communication. Usually the copyright holder can only get grip of one of the middle-men of the communication path, instead of the real peers exchanging data.

To a certain extend, the later position can also be taken by Bob, in case he has a reason trying to identify Alice. This can be, for example, in the case of blackmailing, or any other crime being carried out over anonymous networks.

All of these entities have in common that without any further means it is next to impossible for them to identify the real sender and recipient of a message – mostly because this is the purpose, why anonymous communication means have been chosen.

There are a number of papers in this research area trying to propose solutions for *conditional* anonymity, i.e., solutions where some communication partners can be identified under certain condition, which can include the existence of an issued warrant or the cooperation of one or several trusted third parties.

18.3.2   Techniques and Approaches

This section is about the theory of anonymization techniques and covers basic algorithms and protocols that are used to achieve the goals.

The basic techniques for provision of network layer anonymity can be categorized as follows:

18.3.2.1   Single Hop Proxies

This is one of the currently most popular and probably easiest methods of anonymization to deploy and analyze. The idea is to hide the relationship between communicating parties by making use of a single proxy server which strips information about the request originator. It is usually applied for anonymization of HTTP requests. For the peer partner it appears as if the request is originated by the proxy server, and not by the user. These kind of

proxies can be used either by configuring a proxy server setting in the web browser or setting instructions to fetch the corresponding web pages from a web interface. In the second case, while delivering the requested document, all the links are rewritten so that they point back to the original site through the proxy server, and not directly to their source.

If the connection to the proxy server is not encrypted, merely a limited protection against the end server (service provider) is provided. If the tunnel to the proxy server is encrypted, no one along the path to the proxy server (e.g. local administrator or ISP) can per se deduce the addresses of communicating parties. However, if no padding on the packet layer is applied – which is the case by the use of standard software – this approach becomes vulnerable to fingerprinting attacks [Ray00].

Additionally, single hop proxies are single point of failures and trust. A user has to trust the proxy operator and the proxy operator alone has all necessary data to de-anonymize involved users. Furthermore, the approach is vulnerable to an attack, where the adversary can observe all traffic entering and leaving the proxy [Ray00]. Probably the most famous example of a practical realisations of such approach is *anonymizer.com*.

18.3.2.2   Layered Encryption Approaches

This approach is more complex and makes use of distributed trust. A typical representative of the approach to send the data using layered encryption schemes is Tor. The Tor network is an overlay network consisting of servers that are called *onion routers* (ORs). Currently there are about 2,000 ORs in the Tor network that are running more or less permanently. Each OR runs on an Internet end-host and maintains TLS connections to many other ORs at every time. To anonymize Internet communications, end-users run an onion proxy (OP) that is listening locally for incoming connections and redirects TCP-streams through the Tor network. To achieve this, the OP constructs *circuits* of encrypted connections through a path of randomly chosen onion routers. A Tor circuit, per default, consists of three individual hops, of which each one only knows which machine has sent him data (predecessor) and to which he is relaying to (successor). The default circuit length of three hops states a reasonable trade-off between security and performance. To avoid that the last node of a path (*exit node*) learns the first (*entry node*), an additional third node (*middle node*) is used.

During circuit creation, Diffie-Hellman key exchanges are used to establish shared symmetric session keys with each of the routers in a path. A proxy encrypts all traffic that is to be sent over a circuit, using these keys in corresponding order. Every hop on the path removes one layer of encryption while relaying the data, so only the exit node knows the actual destination of a stream. Application data is generally transferred unencrypted on the link from the exit node to the destined Internet end-host, unless an encrypted connection is used, e.g. when using TLS/SSL.

Once a circuit is established, the onion proxy can use it as a tunnel for arbitrary TCP connections through the Tor network, while many TCP streams can share a single circuit. Proxies stop using a specific circuit after a configured amount of time (or data volume), which prevents the users from certain profiling attacks. On the application layer, the SOCKS protocol is used to tunnel arbitrary TCP traffic through the Tor network. For web browsing it is further recommended to point a web browser to Privoxy [Prib], which can be configured to use SOCKS for sending HTTP traffic over Tor while performing basic application layer filtering.

Practical usage of Tor often leads to delays that are not tolerated by the average end-user, which, in return, discourages many of them from the use of the system. The latter indirectly lowers the protection for the remaining users due to a smaller user base. Within the PRIME project we proposed new methods of path selection for performance-improved onion routing that are based on actively-measured latencies and estimations of available link-wise capacities using passive observations of throughput. We evaluated the proposed methods and also present a practical approach to empirically analyse the strength of anonymity, that certain methods of path selection can provide in comparison to each other. Beside the legacy Tor software[9], two additional independent client implementations of the Tor protocol exist. One of them is OnionCoffee[10], where the primary goal was to provide network layer anonymity to be used within the EU Project PRIME.

The OnionCoffee Tor client is implemented in Java and has special emphasis on QoS and user-friendliness. It is additionally equipped with GeoIP data which allows to determine the country and continent a router is located in, while selecting the nodes. This makes it possible to put additional geographical constraints on the circuits. It is currently already possible to, for example, exclude nodes in specific countries from being used in circuits, allow only at most one node from the same country, etc.

The Java Anon Proxy AKA JAP or WebMixes[11] is a different network which is also based on onion routing. One of the main differences to Tor is that the user cannot choose freely a route between the relays. In JAP the mixes are forming pre-built cascades, where the user only has to choose one of the sets.

Tarzan [FM02] and MorphMix [RP02] are two other approaches utilizing onion routing in a P2P manner. In contrast to all other approaches, MorphMix neither requires nor strives to have knowledge about all the nodes in the network. For the circuit setup so-called *"witness"* nodes are used, that facilitate the selection of nodes for circuit extension. Collusion detection mechanisms are used in order to detect "witnesses" that behave unfair offering colluded nodes for selection. Detection is based on the fact, that colluding entities

---

[9] http://tor.eff.org/

[10] http://onioncoffee.sourceforge.net/

[11] Nowadays the commercial version of the system known as Jondonym.

behave differently which can be pinpointed on the long run. However, this protection can be bypassed [TB06]. In Tarzan every node has a set of peers for exchanging cover traffic. These are so-called *"mimics"*. Nodes select their mimics in a pseudo-random universally verifiable way. To achieve this, each peer needs to know all nodes in the system. Circuits are built only through nodes that exchange cover traffic between themselves.

I2P[12] makes use of garlic encryption. The paths are not necessarily symmetric, so different tunnels for in and out-going traffic can be used. Even in one direction packets can take different routes. Currently, like Tor, it is routing with best effort service, but pooling and mixing functionalities are also planned.

Within the PRIME project, a new approach for anonymization that makes use of layered encryptions, was developed. It is motivated as follows. Currently existing techniques are characterized by high complexity of the protocols. This has several drawbacks. First of all, the development effort for practical implementations is rather high. This leads to existence of at most a single implementation. Thus, the risk of having software monocultures becomes very high. Therefore, failure in a single implementation can paralyze/destroy the whole network. Current anonymization protocols are proprietary and not standardized. Existence of additional implementations is further hardened by the fact of changing protocol specification in regular time intervals, like, for example, in Tor.

Implementations of complex systems are more prone to failures than those of simple systems. Another point is the difficulty to analyze properties of the complex system. Thus, formal security analysis becomes infeasible and most disadvantages/attacks on the approach become known only after the network is developed and operated.

Further, only experts know how such complex systems really work, while regular users have only vague knowledge about the actual functionality.

In short, complexity kills security, thus also anonymity.

Available practical implementations suffer from poor performance [PPR08]. This results in a decrease of the user numbers, as users are known to be impatient and not willing to wait for a longer time in order to get a requested web page.

We introduced a novel lightweight approach for anonymization named Shallon. It is based on standardized protocols, is highly efficient and can be easily deployed. Due to its lightweightness, the properties analysis of the protocol becomes simpler. Because the design is based on the standardized protocols, the deployment becomes trivial and the vision about availability of multiple implementations becomes viable. Shallon makes use of an onion-encrypted HTTP-based tunneling method. This is done with the HTTP CONNECT method/command. The CONNECT method extends a connection in the following way: it instructs the HTTP server or proxy to make a TCP

---

[12] http://www.i2p2.de/

connection to some specified server and port, and relay the data transparently back and forth between that connection and the client connection. After extending the connection from one proxy to the next, an SSL handshake is performed with the new node in order to retain confidentiality of the next hop and/or application layer data. Therewith, the messages on the path are encrypted in an onion-like manner.

18.3.2.3   Simple Randomized Routing Protocol

Crowds [RR98] was designed as an alternative to the techniques in the previous sections. It is based on a simple randomized routing protocol, where all participants forward messages on behalf of other users as well as their own. The main idea of Crowds is to hide each user's communications by routing them randomly within a group of similar users ("blending into a crowd"). When a user requests a web page, the request is sent to another (randomly chosen) crowd member. This member decides whether to forward the message to its final destination or to some other random participant making a biased coin toss. Communication between nodes is encrypted, however each of them sees the content of passing messages, including the address of the final destination.

GNUNet [BG03] also makes use of a simple randomized routing, where the forwarding on behalf of the others (the so-called *"indirection"*) depends, among other things, on the network load.

18.3.2.4   Multi- or Broadcast Based Methods

The first proposal based on broadcast techniques was a DC-network as defined in [Cha88]. It can provide perfect anonymity, however under some rather demanding assumptions. It requires all nodes to communicate with each other for every message transfer, thus it requires secure and reliable broadcast channels, is prone to channel jamming, inefficient in large networks, etc [Ray00].

P5 [SBS02a] is another approach from this category which is more scalable because of the network division into a tree hierarchy for smaller broadcast groups.

Finally it has also been shown how to use Satellite ISPs for anonymous data communication [AG07].

18.3.2.5   Censorship Resistance

Censorship resistance deals with an attempt to prevent censors from the acquaintance of distribution of a particular content through the network. Providing resistance against censoring is a very challenging and difficult task to achieve. However, it is vital for the purpose of freedom of speech, mind and achievement of democratic principles in today's society.

According to the Universal Declaration of Human Rights, everyone has the right to freedom of opinion and expression, including receiving and imparting information and ideas through any media regardless of frontiers [UDH98]. In todays world, however, an increasing number of organizations, companies and even countries block the free access to parts of the Internet [UDH03]. The censors try to impede accessing some special political, ethical or religious content. For example, Saudi Arabia runs a country-wide Internet Service Unit (all ISPs must, by law, route through it), which provides an infamous web-censoring system that is supposed to protect Saudi citizens from "those pages of an offensive or harmful nature to the society, and which violate the tenants of the Islamic religion or societal norms"[13]. Another well-known example is the "Great Firewall of China", where strict censoring is provided at the governmental level. Lots of web pages like the British radio station BBC, human rights organizations, or the free encyclopedia Wikipedia are blocked. According to an Amnesty International report, there are 54 people in jail in China because of illegal content distribution[14]. International Internet search engines like Google, Yahoo and Microsoft's MSN were recently criticized for censoring search results according to China's guidelines. Moreover, content filtering is also a subject in democratic nations. So, for example, US Marines Corps censors web access for troops in Iraq[15,16]. The European Union considers filtering and ranking according to the Internet Action Plan [EU006].

For the purpose of freedom of speech, mind and achievement of democratic principles there is a great demand to withstand filtering and censoring of information access and dissemination. Blocking resistant[17] systems try to provide as much reachability and availability as possible, even to users in countries where the free flow of information is organizationally or physically restricted [KH04].

Censorship resistant systems often have to provide anonymity to its users in order to grant their protection (especially from the blocker) and therewith to achieve desired properties of the system. Providing resistance usually requires distributed, peer-to-peer systems in order to overcome the blocking of the central server entity. Distributing functionality across many network nodes allows to avoid an obvious single point of failure where an attacker can clog the entire network. Using peer-to-peer based systems, though, requires the need to place trust on peers in the network. For this purpose reputation can be introduced. However, if the main objective of the network is to provide support for anonymity, the realization of the reputation itself becomes very problematic. Hiding the real identity gives a possibility for an attacker to easily throw away a pseudonym that has acquired a bad reputation. Furthermore, it is difficult

---

[13] http://www.newsforge.com/article.pl?sid=04/01/12/2147220

[14] March 2006, see also http://www.heise.de/newsticker/meldung/70800

[15] http://yro.slashdot.org/article.pl?sid=06/03/07/1613236

[16] http://wonkette.com/politics/wonkette/our-boys-need-gossip-158687.php

[17] We use terms "blocking resistance" and "censorship resistance" as synonyms.

to verify a member's behavior while keeping his status anonymous as these are two contradictory things. However, to the favour of blocking resistance, blockers and "normal" users have different objectives which can serve as an incentive for the classification.

Within the research work in PRIME [PP07, PP06b] we have defined our model of a censorship resistant system and proposed to split the problem into a net of trust and steganographic data transfer. Steganographic communication is necessary to hide the traffic to and from the system as well as between the users. The net of trust is needed in order to find peers for communication and prolong contacts among them. We have proposed to realize it as a collusion-resistant, probabilistic directory. A definition of a set of properties has been given that this directory must fulfill. With the simulation-based evaluation we have shown that clustering users based on their trust is a very promising method to build a directory with the before mentioned properties. We achieve this by clustering the system users into disjoint sets, instead of calculating a global value of trustworthiness.

To ease the implementation, we have investigated the approach of a centralized directory. In order to provide protection against denial-of-service attack, single point of failures and, not less important, to make it difficult to block the access to the central entity, switching to a distributed directory and its implications must be researched and implemented.

All in all it is hard to say at this point to which extent our results are applicable to real systems. Even though we took care to choose a powerful social model, it is very difficult to sufficiently abstract and simulate the human behavior and interpersonal trust. Therefore, in order to make final conclusion statements about our approach, evaluation in real-world settings are necessary.

18.3.2.6   Strong Primitives

In order to obtain strong anonymity against global observers, different primitives can be used: superposed sending allows to be unobservable while sending, encrypted padding while sending or receiving, and broadcast with implicit addresses and private information retrieval allow to be unobservable while receiving.

Superposed Sending

Description. Superposed sending as an anonymous communication tool was first introduced by David Chaum in [Cha88]. The basic idea was presented through an allegory of three cryptographers on a dinner wishing to know if one of them had paid the dinner, without knowing whom (i.e., they wanted to be able to say "I have paid" with sender unobservability). Computer networks that implement the resulting protocol are called dining cryptographers networks (DC-nets); the protocol itself is named the DC-net protocol.

Most of the literature on superposed sending is dedicated to the detection of disrupters [WP90, Wai90, BdB90, vABH03, YF04, GJ04] (i.e., system users that disrupt other users' communications), but recently two papers [DO00, SBS02b] studied how to implement better superposed sending protocols, focusing in key distribution and the use of multiple-level hierarchical topologies. However, none of these protocols (nor the original one) have been implemented, except for experimental purposes.

In a superposed sending protocol, all the participating users send scrambled messages at each round, even if they don't have anything to transmit. The protocol described below shows how a DC-net round is performed.

Protocol. DC-net Round.

Let $U_1, ..., U_n$ be a set of users such that each couple of users (U_i, U_j) shares a unique common l-bits long secret, $S_{i,j} = S_{j,i}$.

Each user U_i encodes a message $Message(i)$ as a string of l zeroed bits if he has nothing to transmit, or as an l-bit message if he has something to transmit.

Round progress :

1. Every user U_i sends $Emission(i) = Scrambling(i) \oplus Message(i)$ with $Scrambling(i) = \bigoplus_{j=1, j \neq i}^{n} S_{i,j}$.

2. The recipient computes the result of the round as $Result = \bigoplus_{i=1}^{n} Emission(i)$.

Since every $S_{i,j}$ appears twice (once inserted by U_i and once inserted by U_j), the scramblings cancel each other and the recipient obtains $Result = \bigoplus_{i=1}^{n} Message(i)$.

The $S_{i,j}$ must be renewed each round. They can be randomly generated and stored in large storage devices (DVDs, for instance), that are exchanged physically by the users, or be pseudo-randomly generated [BK06] from a secret seed known only to users U_i and U_j.

Collisions. In a given round, if only one user has attempted to transmit, the result of the round is his message (as all the other messages are composed of zeroed bits). If more than one user has attempted to transmit, there is a *collision*. The easiest way to deal with a collision is to wait for a random number of rounds before trying to transmit again, but there are more efficient ways to deal with them. In particular, superposed receiving and channel reservation provide simple solutions to this issue.

Superposed receiving [Pfi90] (see [Wai90] for a reference in English) allows to solve a collision of s messages in s rounds. Superposed receiving has a very small communication and computing overhead, and is therefore a much better solution than waiting for a random number of rounds if collisions are frequent.

In [MD06] we proposed the use of multiple independent superposed sending channels when dealing with low-latency and communication-oriented systems,

Fig. 18.8 Channel reservation

and called this approach channel reservation. This technique consists in having a control channel where there is a superposed sending round with a given frequency (for instance, every few seconds). When a user wants to begin a communication he first sends a message using the superposed sending protocol through the control channel, asking for the creation of a communication channel. When the result of a round on the control channel reveals a channel creation request, all the users begin an independent set of superposed sending rounds at communication frequency. We call this independent set a communication channel. In this channel only transmits the user who has requested its creation, and therefore no collision occurs. When he finishes his transmission, he sends through the communication channel an agreed message for channel termination and users stop the rounds associated to it (see Fig. 18.8). Collisions may occur in the control channel, but as channel creation requests are generally much scarcer than message sending, the number of collisions to be resolved is greatly reduced.

Unobservability properties. A set of users transmitting through a superposed sending protocol form a sender unobservability set against any attacker, even the recipients of the message. If an attacker controls a subset of users, the non-controlled users continue to form a sender unobservability set against him, whatever the size of the subset is.

Performance issues. If the users participating in the superposed sending rounds are distributed over the Internet, there are serious performance issues, since all of the users' messages are needed to obtain the result of a superposed sending round. Today's Internet connections have good mean throughput and latency, however the performances are very variable from one connection to another and even at different instants for a given connection. With a superposed sending protocol, the latency of a round is always larger than the largest of the users' latencies and the throughput lower than the lowest of the users' throughputs. This protocol should therefore be used over the Internet for high-latency and low-throughput communication only.

In a LAN, the user connections have stable enough throughput and latency and thus this protocol can be used to transmit a VoIP communication flow. However, the maximum number of users is limited. Indeed, if a single user is unable to participate to a given round, the result of the round is scrambled

scrambled. Even if all the users can guarantee a 99% uptime, when there is more than one hundred users participating in a superposed sending protocol, scrambled rounds will be very frequent. For example, for one hundred user with a one percent probability per round of failing to participate to the superposed round (which corresponds to the 99% uptime assumption), the probability for a given round to be scrambled is over 63%. With strong latency constraints this issue is amplified, as not only the downtime of users is critical but also any delay exceeding the acceptable RTT (which at least would result in jitter, or more probably on the server dropping the round).

Encrypted Padding

Description. If a user A sends directly messages to another user B, an attacker eavesdropping on the link between A and B learns how many messages have been sent, when, and the size of each message (even if they are encrypted). To hide this information, user A can use *full encrypted padding*, that is, send to B fixed-size encrypted messages every τ seconds. Each of these messages' associated cleartext is garbage as long as A has no information to send to B. When A wants to send a message to B she encapsulates it inside the encrypted padding messages replacing the garbage with the information to be sent. B decrypts all the messages received from A. As long as the resulting cleartexts are garbage B dumps them. When the resulting cleartext contains useful information (which can be revealed by a given format or marker), he reads it.

The basic idea is that, if we consider that it is not possible to distinguish between the encrypted messages containing garbage and the encrypted messages actually containing information for B, an attacker always sees the same thing: a constant rate flow of fixed-size encrypted messages between A and B.

Unobservability properties. If an attacker cannot distinguish encrypted padding from the other encrypted messages, A forms a completely unobservable sender singleton. Of course, A's unobservability cannot hold against B, as he can decrypt the messages. Similarly, if the attacker is not be able to decide whether B receives encrypted padding or not, B forms a completely unobservable recipient singleton (except with respect to A).
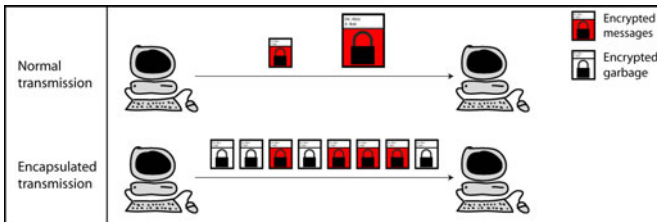
Fig. 18.9 Encrypted padding

Randomized public key cryptosystems with the security property of *indistinguishability*, such as [Pai99], have formal proofs that attackers are unable to distinguish between the numerous[18] encryptions of any two encrypted messages. In particular, this implies that the attacker is unable to distinguish between encrypted padding and encrypted messages containing useful information. Using a public key cryptosystem is costly so usually it is accepted that a strong symmetric encryption system such as AES can be used as long as some randomization is introduced.

There are two evident limitations to the users unobservability. First, if A wants to be unobservable while sending messages to another user C, she must have another encrypted padding channel with him. Similarly, B must have an encrypted padding channel between C and him if he wants to be unobservable while receiving messages from him. Second, if the encryption padding has a given throughput, let's say 10 Kbits/s, A cannot send information to B at a higher rate without being observable.

Alternatives to the full padding approach exist. For example, A can send encrypted padding following a random distribution instead of sending them at a regular pace. To simplify the presentation of the different techniques do not discuss these approaches, letting the reader infer from the contents of this paper how they could be used replacing full encryption padding.

Broadcast with Implicit Addresses

Description. Sending messages that everybody receives (or can receive) such that only the real recipient is able to decrypt them is a classical mean to ensure recipient unobservability. For example, coded messages were broadcasted by radio during World War II to the resistance. Of course, nobody but the recipients could say which radio listeners were able to decrypt the messages and which not and therefore recipients were unobservable. A similar situation is found in spy movies with coded messages on newspapers, everybody receives them, and there is no way to know who understands what they mean and who does not. On a computer network, broadcast allows to send a message to all the addresses of a given network or sub-network. Its usage is however constraining as the communication links of all users are encumbered, and even if we can broadcast in WANs it is not possible to do it at large scale (for example over the whole Internet).

When users receive a broadcasted message they must be able to distinguish whether they are the intended recipient or not. The easier way to implement this, as in World War II radio broadcasts, is for each user to try to decrypt the message and conclude, depending on whether he is able to decrypt it to a meaningful cleartext or not. Setting whether a user is the intended recipient or not using this approach is called implicit addressing. To simplify the process of distinguishing meaningful and meaningless cleartexts, messages can of course

---

[18] Indeed, in a randomized public key cryptosystem, to each cleartext corresponds a huge set of different cyphertexts.

be formatted in a particular way or contain a tag indicating it has been correctly decrypted.

Unobservability properties. When a message is broadcasted with an implicit address to a set of users, they form a recipient unobservability set against any attacker except the creator of the message who generally knows which user is able to decrypt the message.[19] If an attacker controls a subset of users, the non-controlled users continue to form a recipient unobservability set against him, whatever the size of the controlled subset is.

Performance issues. Decrypting all the messages of all the users which are broadcasting with implicit addresses can quickly become computationally unaffordable. In a communication-oriented context this computational cost can be drastically reduced. Indeed, all the packets of a communication have the same recipient. When a user starts broadcasting a communication with an implicit address, all the users attained by the broadcast will just try to decrypt the first message of the communication, and if they do not succeed, they will infer that they are not the recipient of the communication and stop trying to decrypt the corresponding messages. Thus, a user will just have to decrypt a message every time a communication starts and not every time a message is sent. If users are unable to know when the communications start (because the sender is using a sender unobservability primitive), the process is a little more complex. The basic idea is to have, besides the communication rounds, a less frequent signaling round in which everything is decrypted. Of course, this implies a tradeoff between interactivity and computational cost (see [MD06] for more details).

Private Information Retrieval

Private Information Retrieval (PIR) is a field of research dissociated from anonymous communication. The first link between PIR and communication systems was done by [CB95] to provide a message service with location privacy for mobile users. In [MD05a] and [MD06] we propose to use PIR protocols as a much more communication-efficient alternative to broadcast with implicit addresses. In these papers it is shown how using these protocols enlarges the spectrum of servers providing anonymous communications and allows to obtain much more efficient servers than with the usage of the classical primitives when the users are distributed over the Internet.

In this section, first we provide a description of the PIR research field and then we describe how to use it for anonymous communication.

---

[19] Note that in some situations it may be possible that a user encrypts and broadcasts a message with a key without being able to know to which user this key is associated and therefore the unobservability property would be held even against the message creator.

Description. Usually, to retrieve an element from a database, a user sends a request pointing out which element he wants to obtain, and the database sends back the requested element. Which element a user is interested in may be an information he would like to keep secret, even from the database administrators. For example, the database may be:

- an electronic library, and which books we read may provide information about our politic or religious beliefs, or details about our personality we may want to keep confidential,
- stock exchange share prices, and the clients may be investors reluctant to divulge which share they are interested in,
- a pharmaceutical database, and some client laboratories wish that nobody may learn which are the active principles they want to use,

To protect his privacy, a user accessing a database may therefore want to retrieve an element without revealing which element he is interested in. A trivial solution is for the user to download the entire database and retrieve locally the element he wants to obtain. This is usually unacceptable if the database is too large (for example, an electronic library), quickly obsolete (for example, stock exchange share prices), or confidential (for example, a pharmaceutical database).

Private Information Retrieval schemes aim to provide the same confidentiality to the user (with regard to the choice of the retrieved element) than downloading the entire database, with sub-linear communication cost. PIR was introduced by Chor, Goldreich, Kushilevitz, and Sudan in 1995 [CGKS95]. In their paper, they proposed a set of schemes to implement PIR through replicated databases, which provide users with information-theoretic security as long as some of the database replicas do not collude against the users.

Here we focus on PIR schemes that do not need the database to be replicated, which are usually called single-database PIR schemes. Users' privacy in these schemes is ensured only against computationally-bounded attackers. It is in fact proved that there exists no information-theoretically secure single-database PIR scheme with sub-linear communication cost [CGKS95]. The first single-database PIR scheme was presented in 1997 by Kushilevitz and Ostrovsky, and since then improved schemes have been proposed by different authors [Ste98, CMS99, Cha04, Lip05, GR05, AMG05].
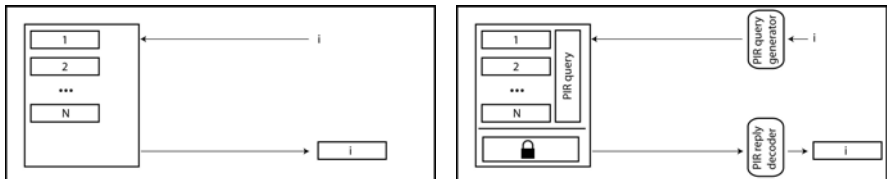
Fig. 18.10 Classical and PIR retrievals

Generally, in a PIR protocol (see Fig. 18.10) a user generates a query (using a randomized algorithm) for the element he wants to retrieve and sends it to the database. The database mixes the PIR query to the database elements (using a deterministic algorithm) and obtains a result which is sent back to the user. The user is able to recover the element he wanted to retrieve out of the PIR reply. User privacy is ensured as obtaining any information about which element was retrieved from the PIR query or the PIR reply implies breaking a well-known cryptosystem (such as Pailler's IND-CPA encryption scheme [Pai99]). Current PIR schemes are very efficient from a communication point of view. In [Lip05], a scheme is proposed in which the user sends a small query and obtains the database element he is interested in with a database reply expansion factor of 2, independently of the number of elements contained on the database. We use this protocol as a reference for our servers.

18.3.2.7   Usage with an Anonymous Communication Server

A pretty important fact about PIR schemes is that in all of the existing protocols the request is independent of the database contents. The same request can therefore be used to generate different replies as the database evolves. Besides, a server providing an anonymous communication service holds the different communication streams sent by the users (whether they use encrypted padding or superposed sending to obtain sender unobservability). This server can be seen as a database, with a slot for each of these streams, that evolves very quickly. If the server generates a PIR reply every time he updates the stream slots, the user doing the PIR query retrieves a communication stream without the database being able to know which stream the user has selected. In figure 18.11 a user has sent a query to retrieve the contents of the second slot of the server (which is receiving three streams). Each time the server updates the slots she generates a PIR reply. The user decodes the replies sent by the server recovering all the messages of the second stream.
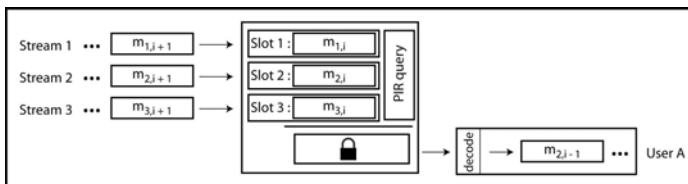
Fig. 18.11 Private stream retrieval

One major issue is the fact that a given user does not know, a priori, whether one of the incoming communications is intended to him or not, nor which. The users must obtain this information somehow while remaining untraceable. See [MD06] for protocols dealing with this issue. In order to focus

on the primitives presented in this section, we just assume that the users are aware of when they receive a communication and which is its index among all the streams the server deals with.

Unobservability properties. If a set of users follows the approach described in the previous section they form a recipient anonymity set even with respect to the server administrator, as he is able to know if a user is receiving a communication or not but unable to learn which communication corresponds to which user. The reason why the server administrator is able to know if a user is receiving a communication is that he is aware of which users send PIR queries and receive a reply stream, and which not. To obtain recipient unobservability, a user must therefore follow either a full padding or a superposed sending approach to send its PIR queries. The full padding approach is described below. The superposed sending approach is described in [MD06] as its relevance is highlighted better then.

If the users follow a full padding approach, each of them sends to the server a PIR query every τ_{switch} seconds. When a user wants to receive a communication he sends PIR queries for the slot corresponding to the communication he is interested in, and when he has nothing to receive he sends queries for a random slot and drops the PIR replies received. As the server is unable to learn anything about the users' choices from the PIR queries or from the PIR replies he generates, the users form a completely unobservable recipient set. The value of τ_{switch} is pretty important as it defines the maximum time a user may have to wait when trying to switch from one communication to another. Imagine a user A learns (no matter how) that he is going to receive a communication with index 2 but he has just sent a fake query for the another index. A has to wait for τ_{switch} seconds before sending another query and being able to receive the communication. If τ_{switch} is too large the delay may be unacceptable. On the other hand, the smaller is τ_{switch} the larger the overhead induced by the fake PIR queries is, and therefore a trade-off must be found for each application. In most VoIP systems a few seconds will be an acceptable value for τ_{switch}.

18.3.3   Threats in Anonymous Communication

This section will cover the most important and imminent dangers in the area of anonymous communication. As opposed to Section 18.3.2, which lists the basic techniques to reach anonymity, this section will list conditions and situations, where the provided degree of protection will fail.

As anonymous communication strives to provide protection on the network layer, there are naturally countermeasures and attacks on the same layer trying to defeat the protection.

The most simple attack is to block access to the network (aka "'denial of service"'-attacks), thereby forcing its user to either stop communicating, or to communicate in plain (and thus revealing their peer partners).

There are several ways of mounting denial of service attacks. Even an external entity, and also a communication peer, can try to shutdown a network by taking down central infrastructure necessary to connect or use the network. Depending on the implementation of the network used that can be either the directory service listing the addresses of the nodes in the network, or the nodes themselves. "Taking down" can be achieved by trivial means like bandwidth exhaustion attacks, i.e. sending a high number of requests to them, routing attacks that change the path of messages, or DNS spoofing or poising with similar effects. Since it is very difficult, if not impossible to protect against these attacks, networks that rely on centralized infrastructure are highly vulnerable to this kind of attack. Fortunately, this has not been yet observed in real networks up to now.

For a local administrator or ISP there are (in addition) more elegant and trivial means of blocking access. Besides blocking access to certain network addresses and (TCP-)ports, an administrator also has the opportunity to look at the content of a user's data streams and decide to redirect or drop connections that look suspicious or similar to a network protocol that is being used in anonymization. These attacks can ultimately only be thwarted by *steganography*, and to a certain extend by embedding the communication either on traffic layer or protocol layer within other communication streams.

A different kind of attack is possible for local administrators or ISPs, which are between the user and the first hops of the network.

Anyone between the user and the first hop of the network, like the local administrator or the ISP, can try to identify the type of traffic anonymized by the user by statistically analysing traffic patterns. This has been extensively done for web traffic, to the extend that attackers could identify certain web pages accessed by the user by characteristics in numbers and delay between single data packets [Hin02].

This research was followed by a work of Serjantov and Sewell which analysed the general properties of hiding connections in anonymizing networks [SS03]. They identified a set of preconditions under which packet counting attacks were feasible for an attacker and could be used to identify individual connections. In addition, they also shortly discussed countermeasures, which however are impractical to deploy in large-scale systems.

A very innovative attack has been presented by Murdoch and Danezis in [MD05b], where a remote attacker can use probing messages through the anonymization network in order to gain information about the path of an arbitrary user through the network. With this attack, which is nearly transparent to the victim, the attacker can trace him down to the first node on the network. If the victim is participating as a node himself, it is even possible to identify himself. This attack reduces the protection of low-latency networks like Tor which should provide a rather high practical level of security to the protection provided by a single proxy hop.

End-to-end timing has been used in [ØS06] to identify IP addresses and identities of location hidden servers in the Tor network, i.e. services which

are designed to provide network services while providing anonymity for the identity of the server. To this end, an attacker builds repeating connections to the hidden service which in return has to start building virtual circuits through the Tor network due to the specifications of the Tor protocol. Deploying a single node in the network, the attacker has then a certain probability for each of these connections to be chosen as first hop, thus learning the true identity of the server providing the hidden service.

One of the traditionally neglected attack vectors is a special case of the partial present attacker. An attacker controlling the exit node of a user's stream out of the network, is able to see most of the user's data in plain. The potential for abusing this position has up to now not been fully researched. We will summarize some of the preliminary findings in the next paragraphs.

Since most of the users of an anonymizing network are using it to hide their identity from their peer partners using traditional network protocols like HTTP and email, the traffic exiting from the network is typically unencrypted standard network protocols. This gives the opportunity for the end node operator to read or change all data exiting from the network.

Passive attacks, i.e. sniffing data from the user like credit card numbers or any other sensitive personal information, will go unnoticed by the user. The attacker can use this in order to run identity theft or impersonation attacks. In order to identify his victim, an attacker can run extensive profiling algorithms on the data provided by the victim itself. This is not only data deliberately, and most often unintentionally, given by the user: the name, address information, a language, or similar. This includes also data in the header fields of communication protocols, like e.g. HTTP.

Coming out of passivity, an attacker highly increases his chances in identifying the victim: a set of techniques that include injecting booby trap-like structures into the application layer was proposed in [For06], and also verified in experiments for effectiveness. The results were devastating with regards to the awareness of the users to attacks through this channel at that time: the success rate was far beyond 90%, allowing to identify an significant amount of the networks users at that time.

Finally, an attacker can even try to intercept encrypted and authenticated sessions, like TLS secured HTTP-sessions. This usually means that the attacker has to replace the certificate of the user's peer with a self generated one and spoof the identity of the original server. Of course, any user remotely aware of security issues will in this case see a software warning that the presented security certificate does not fit to the address that the user entered.

In addition, it has been shown by Murdoch in [MZ07] that controlling a single central hub of the Internet is sufficient to deploy timing and correlation attacks. This even is true, if the attacker is able to intercept only a fraction of all traffic running through the hub, like one out of 10,000 data packets.

18.3.4   Legal Issues

This section covers some major problems in the area of anonymous communication. The reader should however be aware that this list is neither complete, nor will it replace by any means a professional law consultancy.

The perhaps most prominent legal issue in this area is the responsibility that node operators bear for forwarding data on behalf of others. As the software is usually designed to remove all hints on the original sender's identity, it is very difficult for the operators to trace a message back to its originator. In case of legal issues however, it is their identity which therewith replaces the original senders identity, thus law enforcement agencies, lawyers, and such, are all contacting them. Depending on the country they life in, they can either just try to ignore legal claims, but most often they have to deal with law enforcement agencies and try to explain, why they are not the actual person which committed the crime. The situation can however escalate, and there have been examples of node operators which had their homes seized and hardware confiscated. Thus, it should be carefully investigated, if one should contribute by running nodes.

In addition to this, operating a node can be difficult due to local laws against strong cryptographic primitives[20].

A very new problem, which has thus not yet been validated in praxis, is the effect of the European data retention directive. A potential impact could lead to either the legal impossibility to use anonymizing networks, or force operators to keep track of relayed messages. The latter however would require so much storage space that probably a large portion of nodes will be shut down.

Finally, in some countries, e.g. in some Arabian countries, all traffic exiting the country has to be routed through a central proxy which then filters out illegal web sites. In these countries the use of software to access anonymizing networks, and therewith circumventing the governmental filter, can be considered a criminal act.

18.4   Unobservable Content Access

In most systems security goals are achieved through a mixture of classical security policies (such as access control lists) and standard cryptographic mechanisms such as encryption, digital signatures, key establishment protocols, and the like. In PRIME more complex policies are considered that take the privacy needs of users into account. Moreover these policies make use of anonymous credentials to provide the user with privacy guarantees that are in the users own sphere of control.

This strategy can be taken one logical step further. By taking the privacy relevant parts of application specific policies out of the control of the service

---

[20] See e.g. http://rechten.uvt.nl/koops/cryptolaw/

provider and making them a part of the system itself. We consider a simple example:

Say that a patent database service provider has a policy of not looking at what patents his clients are looking at, and keeping this information maximally secure. The conventional approach would be to collect all kind of personal information, such as credit card numbers, identity certificates, email addresses, name, address, and then provide the service to the subscribed users, relying on the security of the backend system to protect the users privacy. This requires a lot of trust into the service provider, and increases the risk of data compromises. The standard PRIME approach is to collect only the information necessary, and give users anonymous access based on an anonymous subscription credential and anonymous communication mechanisms. Obviously this reduces the reliance on the security of the service provider's system, by allowing users to remain anonymous.

However it is not yet the best we can do.[21] What about taking the privacy promise of the service provider literally? The service provider promises to not record what patents his clients are interested in. Under the standard PRIME approach a compromised service provider can still count the number of (anonymous) accesses to individual patents, and can try to deanonymize users based on traffic or timing information. Using cryptographic techniques like private information retrieval and oblivious transfer it is possible to implement such a service in a way that all the service provider learns is that some patent was accessed, but not which. Solutions for services that protect the privacy in this strong sense are the topic of this section. While these solutions are not yet implemented as part of the PRIME integrated prototype, they play an important role in the foundation of a sound theory about efficient privacy friendly services.

Structure

In the following, we want to give an intuition about how oblivious transfer (OT) can be implemented by a cryptographic protocol. We discuss how a weaker primitive called private information retrieval (PIR) that protects only the privacy of the user but not the secrecy of the database can be combined with oblivious transfer to improve the overall efficiency of the system. As a further step towards the practicality of such services we point out that it is possible to overcome the following seemingly paradoxical problem: OT allows a user to access information without revealing which information is accessed. If this is the case how can the organization enforce different access control or service provisioning policies on its data? Based on this preliminary explanation we disuss how techniques for unobservable content access can be used

---

[21] As noted in Section 18.1 privacy and anonymity degrades if systems are used over a longer time period. Even if the wilful release of application data could be contained, there would still remain the danger of a strong attacker breaking the anonymization at the communication layer as described in Section 18.3.3.

to protect the location information of users of location-based services in a way that is approaching a practical implementation on todays mobile platforms. We conclude by looking at unobservable services from a wider PRIME perspective.

18.4.1   Private Information Retrieval and Oblivious Transfer

Private information retrieval was already discussed in Section 18.3 as a means of achieving anonymous communication. Indeed unobservable content access can be seen as a form of communication, where the sender himself (the database service provider) does not get to know to which user he is sending which (application specific) information.

The privacy requirements of the user imply the need for private information retrieval (PIR) [CKGS98]. However we also want to protect the service provider from leaking all his database at once. Symmetric PIR (SPIR) is required if a service provider wants to avoid leakage of database information that has not been queried. It was shown in [CMO00] that for the case where there is only one copy of the database there exists a communication-efficient reduction from any PIR protocol to a 1-out-of-n oblivious transfer (OT). Moreover for the single copy case SPIR corresponds to 1-out-of-n OT (OT_n^1).

Oblivious transfer was first introduced by Rabin [Rab81]. It captures the on first sight paradoxical notion of a protocol by which a sender sends some information to the receiver, but remains oblivious as to what is sent. The paradox is resolved by recognizing that it are the actions of the *receiver and the sender* that determine the outcome of the protocol. Even *et al.* [EGL85] generalized it to 1-out-of-2 oblivious transfer (OT_2^1). The receiver determines which message out of two possible messages she is going to receive. In turn it was shown how to construct OT_n^1 from n [BCR87] and even $\log n$ [NP99a] applications of OT_2^1. [NP01, AIR01, Kal05] provided direct constructions for OT_n^1 based on the decision Diffie-Hellman assumption and the quadratic residuosity assumptions.

Adaptive OT

For unobservable service access, we are not so much interested in single executions of oblivious transfer, but want to query the same database multiple times at different indexes. This can be achieved by letting the sender commit to the database and running OT_n^1 multiple times. However this is not the most efficient solution. Moreover the security requirements of such a system differ from those of normal oblivious transfer, as the protocol keeps internal state and queries can be chosen adaptively based on the results of previous queries. The first adaptive oblivious transfer protocol was proposed in [NP99b]. Recently more efficient schemes were proposed by [OK04, CT05]. [CNS07] recognized that the last two schemes are based on a common principle to construct adaptive oblivious transfer from unique blind signature schemes.

We briefly sketch the basic idea of the scheme using an example based on Chaum blind signatures (cf. Fig. 18.12). First, all messages are symmetrically encrypted using the RSA signature of the index. $H(.)$ is a full domain cryptographic hash function. The encrypted database $C_1, \ldots, C_n$ is transferred to Alice. When Alice wants to obtain the information for location σ, she runs a Chaum blind signature protocol with the sender to obtain the key (cf. [CNS07]).

Sender   Chooser
$(d,e) \overset{\$}{\leftarrow} \mathsf{Kg}$
For $i = 1 \ldots n$
$\quad C_i \leftarrow \mathrm{Enc}(m_i; H(i)^d)$

$$\xrightarrow{\hspace{1em} C_1, \ldots, C_n \hspace{1em}}$$
$$\xleftarrow{\hspace{1em} H(\sigma)b^e \hspace{1em}} \quad b \overset{\$}{\leftarrow} Random$$
$$\xrightarrow{\hspace{1em} H(\sigma)^d b \hspace{1em}}$$
$$m_\sigma \leftarrow \mathrm{Dec}(C_\sigma; H(\sigma)^d)$$
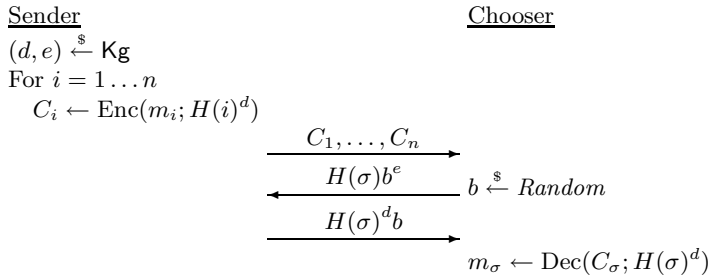
Fig. 18.12 Adaptive OT based on Chaum blind signatures

PIR and Adaptive OT

As already noted in the beginning there is a fundamental difference in the level of protection provided for the database holder between PIR and OT. PIR tries to be maximally communication efficient, but only protects the privacy of the user. For instance, even if the total number of bits retrieved using a PIR protocol is limited, the user may obtain the XOR of different database entries. On the other hand adaptive OT is expensive in terms of communication, i.e., it requires the full encrypted database to be made available to the user, however it ensures that only one database element is retrieved in each transaction. Fortunately, the two approaches can be favorably combined. The encrypted database for the OT can be made available over a PIR server, which improves the communication efficiency of the OT, while the properties of the OT assure database secrecy. Communication efficient commitments to the PIR database, such as Merkle hash trees [Mer88] can be used to assure that the information on the PIR server is not changed between OT invocations.

In turn we will show how OT protocols can be extended to implement access control policies that depend both on attributes of the user and on attributes of the data item retrieved.

18.4.2 Access Control for Unobservable Services

In order to extend OT with access control, the user has to give the database a hidden handle to her choice. The user can create such a handle by committing

to the database index she is interested in. The *hiding property* of the commitment assures that the database does not learn about the users choice, while the *binding property* and the OT protocol guarantee that the user cannot retrieve a different index.

The interface for such a committed OT consists of setup algorithms $I_S(m_1, \ldots, m_n)$ and $I_R()$, as well as transfer algorithms $T_S(state_S, comm)$ and $T_R(state_R, \sigma, open)$ on the sender side (S) and receiver side (R) respectively. Here $comm = \text{Commit}(\sigma, open)$ is the commitment to the users choice.

Now it is possible to do access control using the selective disclosure certification framework from Section 10.3. For instance we would like to impose age restrictions on certain contents. First we associate a minimum age to each database element m_i, $1 \le i \le n$. The database creates a selective disclosure certificate that binds each index to its minimum age: $Cert_1 = \text{Sign}(1, 11; SK_{DB})$, $Cert_2 = \text{Sign}(2, 18; SK_{DB}), \ldots, Cert_n = \text{Sign}(n, 18; SK_{DB})$. The first database element has a minimum age of 11 years, the second and the last of 18 years.

$$
\begin{aligned}
\text{PK}\{(&Cert_{cred}, Cert_i, ID, age, age', i, open) : \\
&\text{VerifySig}(Cert_{cred}, ID, age; VK) = 1 \\
&\land \text{VerifySig}(Cert_i, i, age'; VK_{DB}) = 1 \\
&\land comm = \text{Commit}(i, open) \land age > age'\}.
\end{aligned} \tag{18.19}
$$

A person that is 16 cannot succeed in executing the above proof for $i = 2$, while she can do so for $i = 1$.

Similar techniques can be used to restrict access to persons have a certain nationality, or work in a specific company. More generally it is possible to implement role based access control, where each database resource is assigned several roles. Persons that are assigned these role in their credential can then access this resource. For instance, if we assign the role 'manager' to data element 1, then only a person with role 'manager' can access this record. This can be implemented with the following role assignments, $Cert_1 = \text{Sign}(1, \text{'manager'}; SK_{DB})$, $Cert_2 = \text{Sign}(2, \text{'employee'}; SK_{DB}), \ldots, Cert_n = \text{Sign}(n, \text{'employee'}; SK_{DB})$, together with the following selective disclosure proof,

$$
\begin{aligned}
\text{PK}\{(&Cert_{cred}, Cert_i, ID, role, i, open) : \\
&\text{VerifySig}(Cert_{cred}, ID, role; VK) = 1 \\
&\land \text{VerifySig}(Cert_i, i, role; VK_{DB}) = 1 \\
&\land comm = \text{Commit}(i, open)\}.
\end{aligned} \tag{18.20}
$$

18.4.3   Location-Based Services

An application scenario within PRIME in which the index into the database is highly privacy sensitive is the provisioning of location-based services. The

index used by the user reveals his location. A user's request for the pollen status in a certain area indicates that with high probability this is her current location.

Moreover, the particular setting involves a third party, the mobile operator $\mathcal{M}$ that often is aware of the user's location. Preventing locaton-based services $\mathcal{L}$ from learning the location of users is a key requirement as $\mathcal{L}$ is less trusted than $\mathcal{M}$ following the business model developed in the PRIME project: Small companies (probably not having a well-established reputation) fulfill the role of $\mathcal{L}$ whereas the role of $\mathcal{M}$ is played by big companies with reputation that can be trusted to use the users' location data only for the agreed purposes. Moreover, todays target cellular communication infrastructures do not allow that the location of the user be hidden from $\mathcal{M}$. However, we do not want the mobile operator $\mathcal{M}$ to gain an unfair advantage from his intermediary role, and thus want to hide the users' service usage profiles from him. $\mathcal{M}$ does not need to know which service a user is interested in, which would for instance reveal that she has a pollen alergy. We model different pollen alergy types as multiple services (which could of course be hosted by the same organizational entity). Hiding the location of the user from the service provider $\mathcal{L}$, and the usage profile from $\mathcal{M}$ (in fact we are also hiding them from $\mathcal{L}$) is a natural application for our unobservable service access techniques. As an added benefit, involving $\mathcal{M}$ in the protocol allows us to overcome some of the performance restrictions that would be particularly irksome for users with restricted mobile devices.

18.4.3.1   Definition

Parties

Our protocol involves a user $\mathcal{U}$ who accesses LBSs over her mobile device. Her goal is to obtain location specific information on topics of her interest. This information is collected and served by service providers $\mathcal{L}_1, \ldots, \mathcal{L}_\ell$. A third party that knows the user's location and stands in a financial relationship with the user acts as a proxy $\mathcal{M}$ between users and services — this could be the mobile operator of the user or an organization associated with it. The proxy is responsible for the security of the location information and assists in the payment transaction. We assume that the number of users connected over a proxy is much higher than the number of services. Finally, we assume the existence of an independent party without any commercial interests: a privacy protection organization $\mathcal{T}$ that can be offline for most of the time. We refer to all parties except users as organizations.

Security and Privacy Requirements

A secure and privacy friendly LBS protocol should protect the assets and interests of all involved parties. The assets that need to be protected are: the

user's location, the user's subscription, the topic specific databases of the $\mathcal{L}_j$, and the payment. We consider the following requirements:

- *Location privacy:* The protocol does not reveal the user's location to the service.
- *Service usage privacy:* Even when the proxy and the LBSs collude, the secrecy of the user's subscription remains protected. This includes message privacy; i.e., only the users can decrypt the messages of services.
- *Database secrecy:* The user and the proxy get no information about the topic specific database of $\mathcal{L}_j$. A user gets only the information for the locations she requested. This property must hold even if the proxy and the user collude.
- *Fairness:* It is guaranteed that either the user receives the expected data for the requested location and the LBS receives his expected payment, or the cheating party can be uniquely identified. In order to preserve service usage privacy, the user reveals the cheating party only to the trustee $\mathcal{T}$.

Protocol Phases

In the *Setup* phase the involved parties generate their keys. During the *Service Update* phase, each service $\mathcal{L}_j$ encrypts its topic specific database and transfers it to the proxy. In the *Subscription* phase a user $\mathcal{U}$ creates an encrypted subscription for a service, sends it to the proxy, and is charged the subscription fee. In the *Data Retrieval* phase the proxy runs a protocol with every service $\mathcal{L}_j$ and obtains an encrypted result. The proxy combines them into a single encrypted result for the user such that she only receives the data of the subscribed service. The fair allocation of the money collected in the subscription phase takes place in the *Settlement* phase under the supervision of the trustee $\mathcal{T}$.

Remarks

The database of a service $\mathcal{L}_j$ is represented as a one-dimensional vector with one element for each location. We assume that the number of locations n is the same for all services. Further, we assume that services only update the whole database at once. In the current version of our protocol a user is only subscribed to a single service. Service usage privacy is guaranteed with respect to the total number of users that subscribed during a subscription period. A subscription period is defined as the time between two settlement phases. Finally, we assume that parties communicate over secure channels and that $\mathcal{M}$, $\mathcal{L}_j$, and $\mathcal{T}$, are able to authenticate communication, and to sign messages using their identity.

18.4.3.2   High-Level Approach and First Sketch

We follow a constructive approach in the description of our protocol. In addition to the adaptive OT described above, we introduce new building blocks

and put them into place to describe their function in the construction. Some of the security requirements can be fulfilled by the functionality provided by individual building blocks; others require a complex interplay between building blocks. As a consequence the mapping from building blocks to the sub-protocols of our solution is not one-to-one. We will sketch the sub protocols (cf. Fig. 18.13) as they get assembled from their building blocks.

Homomorphic Encryption

Homomorphic encryption is a form of malleable encryption. Given two cipher-texts, it is possible to create a third ciphertext, with a plain text that is related to the first two. For (additive) homomorphic encryptions, the encrypted plain texts fulfill the following relations:

$$\text{Enc}_h(m_1) \oplus \text{Enc}_h(m_2) = \text{Enc}_h(m_1 + m_2), \qquad c \otimes \text{Enc}_h(m) = \text{Enc}_h(c \cdot m).$$

We speak of additive homomorphic encryption because $+$ corresponds to the addition operation of a ring. We write $c \otimes \text{Enc}_h(m)$ to denote the c times homomorphic addition of $\text{Enc}_h(m)$. Note that for Damgård-Jurik Encryption [DJ01] $c \otimes \text{Enc}_h(m)$ corresponds to $\text{Enc}_h(m)^c$ and can be implemented efficiently.

OT Using Homomorphic Encryption

It is a known property of additive homomorphic encryption that given an encryption $Q = \text{Enc}_h(1)$ it is possible to compute an encryption of a message m as $m \otimes Q = \text{Enc}_h(m \cdot 1) = \text{Enc}_h(m)$. However, if $Q = \text{Enc}_h(0)$, the same operation does not change anything, i.e., $m \otimes \text{Enc}_h(0) = \text{Enc}_h(0)$ [OS05].

Given the semantic security of the encryption, the party trying to encode the message cannot distinguish the two cases above. Based on this observation an OT scheme can be constructed by using a vector $\mathbf{Q} = (Q_1, \ldots, Q_\ell)$. To request message $m_{\hat{\jmath}}$, $Q_{\hat{\jmath}} = \text{Enc}_h(1)$ and $Q_j = \text{Enc}_h(0)$ for $j \neq \hat{\jmath}$. Zero-knowledge proofs can be used to prove the correct construction of Q. The communication complexity of the protocol can be reduced by computing $E = \bigoplus_{j=1}^{\ell} m_j \otimes Q_j$, and transferring only E to the recipient.

Threshold Encryption

In a distributed decryption protocol a private key is shared among a group of parties, where only a qualified subset of the parties is allowed to decrypt a ciphertext c, whereas fewer parties learn nothing on the secret nor on the decryption of c. In our scheme we use the special case of a distributed 3-out-of-3 threshold encryption scheme, which could be implemented, e.g., with the threshold protocol presented in [DJ01].

Zero-Knowledge Proofs of Knowledge

A *zero-knowledge* proof is an interactive proof in which the verifier learns nothing besides the fact that the statement proven is true. Zero-knowledge proofs-of-knowledge protocols exist for proving various statements about discrete logarithms in groups of known and hidden order [BCM05, Bra97, CS97b, Sch91]. These techniques allow to prove statements about cryptographic primitives that operate in these groups, for instance that two commitments contain the same value, or that a value was verifiably encrypted. Given a statement $\mathsf{Alg}(x) = y$ and $\mathsf{Alg}'(x') = y'$ about two algorithms, with secret input x, x' and public output y, y', it is possible to prove AND and OR relations of these statements. Such protocols can be made non-interactive by applying a cryptographic trick called the Fiat-Shamir heuristic [FS86]. We write in a short form notation, e.g., for AND

$$\pi = PK\{(x, x') : \mathsf{Alg}(x) = y \wedge \mathsf{Alg}'(x') = y'\}.$$

Our main building blocks are the two variants of OT and a threshold encryption scheme. Homomorphic encryption and zero-knowledge protocols serve as sub - building blocks in the previous schemes, but are also used to glue them together in a secure way. The two OT protocols are specifically selected for their good performance under repetition of input data. The blind signature based OT scheme is optimized for the case that the input database remains fixed, while the index varies. The homomorphic encryption based OT is efficient in the opposite case; it is efficient for fixed indices.

During the protocol execution, a single proxy interacts with a multitude of users and multiple services. The first building block we put into place is a blind signature based OT protocol. It is executed with the proxy acting as the requester and one of the services as the sender. It allows the proxy to retrieve location specific information $m_{\hat{\imath},j}$ for a user at location $\hat{\imath}$ without service L_j learning the user's location. This guarantees *location privacy*. The proxy executes this sub-protocol with all offered services. This assures *service privacy* at the service side. In this way the proxy obtains an information vector $m_{\hat{\imath},1}, \ldots, m_{\hat{\imath},\ell}$.

Our second building block is a homomorphic encryption based OT protocol. It is run with the proxy acting as the sender (using the aforementioned vector as input) and the user acting as the requester (using the index of the service $\mathcal{L}_{\hat{\jmath}}$ she subscribed to as input). The protocol allows the user to learn $m_{\hat{\imath},\hat{\jmath}}$ without the proxy learning the user's subscription; we achieve full *service privacy*.

Note how the choice of OT protocols is crucial for the performance of our protocol. In the first OT, the same database is queried by the proxy for all users (and different locations as they move about). The database needs to be encrypted and transferred to the proxy only once (cf. Fig. 18.13.2). For the second OT between user and proxy, the subscribed service is invariant for the duration of a subscription period and it is sufficient to send the first (and

expensive) message of the homomorphic OT only once (cf. Fig. 18.13.3 ①).
Consequently we split off these operations as sub-protocols which have the
semantic of a service update and a user's subscription.

This gives us a first instantiation of the first 4 protocol phases. The outline
of the protocol is depicted in Fig. 18.13. Note that some of the sub protocols
are not yet implemented. For ease of presentation we use a simplified notation.
The detailed protocol description is given in [KFF+07].

Setup

(cf. Fig. 18.13.1: ① KeygenU, ② KeygenL) Every user generates a key-pair for
a homomorphic encryption scheme ①. These keys are used for the OT based
on homomorphic encryption. Every service generates a key-pair (skB, pkB)
that is used for OT based on blind signatures ②.

Service Update

(cf. Fig. 18.13.2: ② EncryptData) The database of the LBS $\mathcal{L}_j$ consists of the
n elements $m_{(1,j)}, \ldots, m_{(n,j)}$ ①. Each of the elements is encrypted with its
own symmetric key $H(k_i)$ that is computed by hashing the signature $k_i =$
$\mathrm{Sign}(i; skB)$ of the index ②. The encrypted database $DB_j = (C_1, \ldots, C_n)$,
with $C_i = \mathrm{Enc}_s(m_i, H(k_i))$ is sent to the proxy ③.

Subscription

(cf. Fig. 18.13.3: ① Subscribe) A user's subscription ① consists of ℓ elements
$S_{(U,1)}, \ldots, S_{(U,\hat{\jmath})}, \ldots, S_{(U,\ell)}$, one for each service ②. Each element contains
a ciphertext Q of the homomorphic encryption scheme. Q decrypts to 1 for
the service $\mathcal{L}_{\hat{\jmath}}$ the user subscribes to and to 0 otherwise. To ensure the secu-
rity of the OT the user proves in zero-knowledge that all $S_{(U,j)}$ are correctly
constructed.

Data Retrieval

(cf. Fig. 18.13.4: ① Request, ② Combine, ④ Decrypt) In the data retrieval
phase a user obtains location-specific data from her subscribed service. The
proxy is involved since he is aware of the user's location and stores the en-
crypted databases of the services. Recall that these databases are encrypted
using hashed signatures as keys. The proxy acts on the user's behalf and can
request decryption of individual items without revealing the location of the
user. To guarantee service usage privacy the proxy has to repeat the following
steps for every service $\mathcal{L}_j$ ①:

The proxy blinds the location $\hat{\imath}$ and sends the blinded value $\mathsf{Blind}(\hat{\imath}; b, pkB)$
to the service. The service replies with the blinded signature $\langle k_{\hat{\imath}} \rangle_{blind}$. The
proxy computes $m_{\hat{\imath},j} = \mathrm{Dec}_s(C_{\hat{\imath}}; H(\mathsf{Unblind}(\langle k_{\hat{\imath}} \rangle_{blind}; b, pkB)))$. This com-
pletes the first OT. The proxy collects $m_{\hat{\imath},1}, \ldots, m_{\hat{\imath},\ell}$ and continues with the

second OT (the user's first message is taken from her subscription). The proxy takes the Q corresponding to $S_{(U,j)}$ and computes $E_j = m_{\hat{\imath},j} \otimes Q$ for all $1 \le j \le \ell$. This corresponds to an encryption of $m_{\hat{\imath},\hat{\jmath}}$ for $\mathcal{L}_{\hat{\jmath}}$ and an encryption of 0 otherwise.

As a last step the proxy combines the E_j by homomorphically adding all the encryptions (not knowing which of them contain the message) ②. This way all encryptions of 0 cancel out. The result is transferred to the user ③. She decrypts E to obtain $m_{\hat{\imath},\hat{\jmath}}$ ④.

The first main flaws of this construction is the fact that the proxy learns the $m_{i,j}$ vector for the locations of all users. This is a compromise of *database secrecy*. The second main flaw is the lack of a fair payment infrastructure.
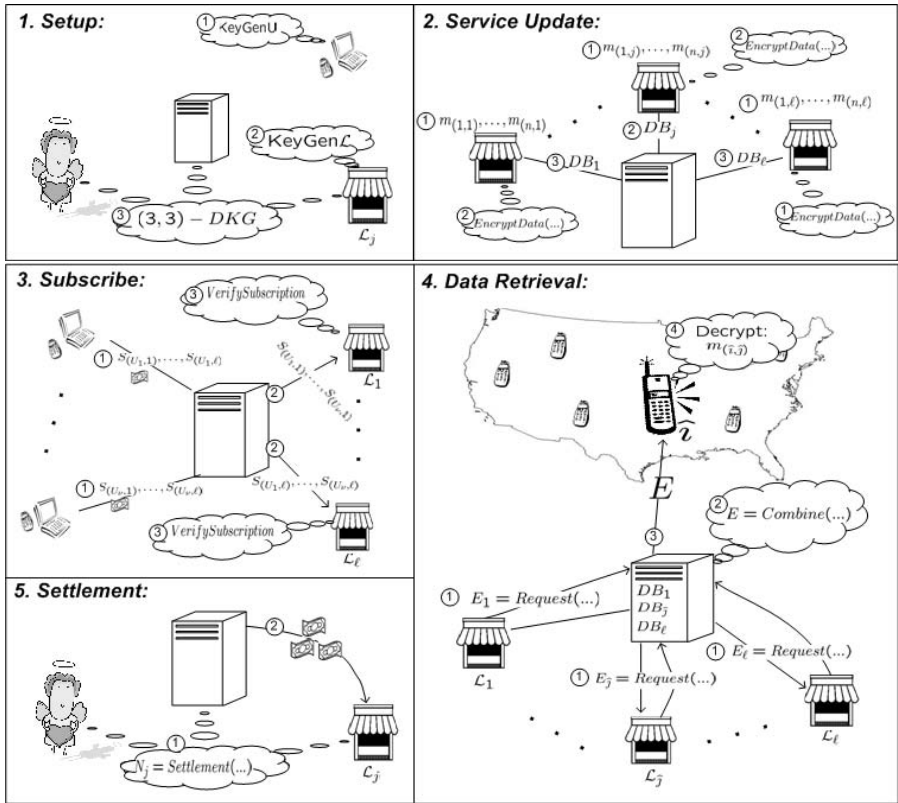
Fig. 18.13 Setup and Service Update, Subscription, Data Transfer, and the Settlement phase: Subscription $S_{(\mathcal{U},j)}$, encrypted database DB_j, service result E_j, combined result E, location-specific message $m_{(\hat{\imath},\hat{\jmath})}$, number of subscriptions N_j, location $\hat{\imath}$, and the subscribed service $\hat{\jmath}$

18.4.3.3   First Revision: Database Secrecy

We address the lack of *database secrecy* by intertwining the first OT with the second. To this end we let the proxy pass on $S_{(U,j)}$ to $\mathcal{L}_j$. Now (after agreeing on who sends which bit range) both $\mathcal{L}_j$ and the proxy can act as senders in the second OT without learning each others inputs. This is made possible by the properties of homomorphic encryption, which lets everyone manipulate encrypted data. Informally, the last message of the first OT will be transferred as part of the encrypted payload of the second OT. This guarantees that only the user with her secret decryption key can obtain the results of both protocols.

   More concretely the following changes have to be made in the subscription and data retrieval phases.

Subscribe

The $S_{(U,j)}$ are now also sent to the services ②.

Data Retrieval

During Request ① the proxy blinds the location $\hat{\imath}$ and sends the blinded value $\mathsf{Blind}(\hat{\imath}; b, pkB)$ to the service. To ensure that only the user (and not the proxy) can decrypt $C_{\hat{\imath}}$, the service encrypts the blinded signature $\langle k_{\hat{\imath}}\rangle_{blind}$. This is done with an additive homomorphic encryption scheme. Remember that during subscription the user (through the proxy) provided the service $\mathcal{L}_{\hat{\jmath}}$ with an encryption $Q = \mathrm{Enc}_h(1)$. The service computes $E_{\hat{\jmath}} = \langle k_{\hat{\imath}}\rangle_{blind} \otimes Q = \mathrm{Enc}_h(\langle k_{\hat{\imath}}\rangle_{blind} \cdot 1) = \mathrm{Enc}_h(\langle k_{\hat{\imath}}\rangle_{blind})$. The result is sent to the proxy who uses a similar approach to add b and $C_{\hat{\imath}}$ to $E_{\hat{\jmath}}$. These requests are done for all services, including those the user did not subscribe to. The latter however received $Q = \mathrm{Enc}_h(0)$ during Subscribe and all the operations result in $E_j = \mathrm{Enc}_h(0)$, for $j \neq \hat{\jmath}$.

   As a last step the proxy computes the homomorphic sum of all encryptions—not knowing which of them contain the unblinding information, the encrypted message, and the blinded signature ②. This way all encryptions of 0 cancel out. The result is transferred to the user ③. She decrypts E, obtains $b\|C_{\hat{\imath}}\|\langle k_{\hat{\imath}}\rangle_{blind}$, and computes $m_{\hat{\imath}\hat{\jmath}} = \mathrm{Dec}_s(C_{\hat{\imath}}; H(\mathsf{Unblind}(\langle k_{\hat{\imath}}\rangle_{blind}; b, pkB)))$ ④.

18.4.3.4   Second Revision: Payment Infrastructure

The core idea for the payment infrastructure is to bind the request of the second OT (the subscription) to a vote. Now revenues can be fairly distributed between services by anonymously counting the number of times users voted for (subscribed to) a service. We use ballot counting techniques based on homomorphic encryption and threshold decryption. We make the following changes to the setup and subscription phase, and we provide an implementation for the settlement phase.

Setup

(cf. Fig. 18.13.1: ③ PaymentSetup) Each LBS $\mathcal{L}_j$ runs a distributed key generation protocol together with the proxy and the privacy trustee ③. This results in a key pair with a secret key shared according to a $(3,3)$-threshold scheme. The shared key is needed in the settlement phase to jointly compute the payment result.

Subscription

(cf. Fig. 18.13.3: ① Subscribe, ③ VerifySubscription) A user's subscription ① consists of ℓ elements $S_{(U,1)}, \ldots, S_{(U,\hat{\jmath})}, \ldots, S_{(U,\ell)}$, one for each service ②. Each element contains two ciphertexts Q and P of the homomorphic encryption scheme, where the first is encrypted with the user's public key and the latter with the payment key. Both Q and P decrypt to 1 for the service $\mathcal{L}_{\hat{\jmath}}$ the user subscribes to, and to 0 otherwise. To ensure the security of the OT and the payment, $\mathcal{U}$ proves in zero-knowledge that Q and P are constructed correctly. The service providers check these proofs before providing the service ③.

Settlement

(cf. Fig. 18.13.5: ① Settlement) The technique used in the Settlement phase is similar to a technique used in electronic voting protocols. The non-interactive zero-knowledge proof sent by the user in the subscription ensures that P and Q encrypt the same value (either 1 or 0). The homomorphic property of the ciphertexts allows to anonymously sum up the content of all different P values. The trustee $\mathcal{T}$ ensures that only the homomorphic sums (and not individual subscriptions) are decrypted in a 3-out-of-3 threshold decryption ①. Based on the result the proxy divides the subscription money received from the users during subscription in a fair way ②.

18.4.4   Conclusion and PRIME Perspective

Privacy solutions can be 'hand-crafted' to fit the specific requirements of individual applications. This is especially relevant for applications that involve sensitive data and a high privacy risk. The tools and techniques used are however of general interest.

19

Reputation Management as an Extension of Future Identity Management

Sandra Steinbrecher, Franziska Pingel, and Andreas Juschka

TU Dresden

19.1 Introduction

Internet users do not only use professional services current identity management assists them in, but more and more also interact with each other or use services created by other Internet users. In interactions with professional services, but even more with unknown individuals, security requirements and trust issues regarding the interaction partner are an important issue. A user has certain expectations on the interaction partners' behaviour which these might fulfil or not. Interaction partners who fulfil these expectations are seen as trustworthy in the future while those who do not seem to be not trustworthy. Most users adapt their behaviour in future interactions to the interaction partners' trustworthiness in former interactions. While professional services try to behave compliant with certain quality standards and legal obligations, such professional service level often cannot be expected from individuals. But this is an issue current identity management does not assist or even address [BPHL+06a, BPHL+07a].

To help new interaction partners to estimate others' behaviour and to motivate interaction partners to fulfil others' expectations, reputation systems have been designed for many applications. They collect the experiences former interaction partners made. These experiences only can give a clue how others might interact in the future because e.g., interaction partners have different expectations, former interaction partners may have lied about others' behaviour [Del00], users may simply fail to fulfil expectations or may suddenly change their behaviour. But despite these uncertainties, a usually large collection of experiences and an honest majority of former interaction partners will

hopefully reach that misbehaviour in future interactions occurs only rarely. Thereby reputation services can be useful for many applications to enhance their users' **trustworthiness**.

A very-popular example of a reputation system is implemented by eBay[1]: eBay offers registered users the opportunity to sell and buy arbitrary items to each other. The exchange of item and money between two users usually is done by bank transfer and conventional mail. Many of these exchanges are successful, but unfortunately some are not. For this reason a reputation system was introduced to collect the experiences sellers and buyers made with these exchanges. After every exchange buyer and seller may give comments or/and marks to each other that are added to the members' reputation (usually together with the annotator and the exchange considered as context information).

But unfortunately, beneath generating trust between interaction partners, the designs of reputation systems currently in use in applications [Kol99] like eBay allow for generating user profiles including all contexts users have been involved in (e.g., time and frequency of participation, valuation of and interest in specific items). These profiles might become a promising target for numerous data collectors. This is contradictory to users' right of informational self-determination [MO04]. This leads to the wish for **privacy-respecting** reputation systems.

The crucial point for the design of a reputation system that is both **privacy-respecting and trustworthy** is that users get at least partial control over the profiles built and users are ensured that others cannot get rid of negative reputation contained in the profiles.

Privacy-respecting design leads to the concept of using pseudonyms in interactions as it is already implemented in identity management and in existing reputation systems. But often in reputation systems others are able to link the pseudonym to a holder. A privacy-respecting design tries to avoid this as long as it is not necessary for an interaction. The linkability might be limited to the provider of the reputation system as in [Del00], to other trusted third parties as in [Vos04, AG06] or to designated identity providers in privacy-enhancing identity management [Ste06, PS08].

The architecture of reputation systems seems to follow a similar path to that of identity management systems: Identity management systems developed from single-application silos that were only used as AAA-infrastructure for one service to stand-alone systems with identity providers allowing propagation of identity and Single-Sign-On. Hopefully the next step will be user-controlled and privacy-enhancing identity management like PRIME will be in common use.

Reputation systems are now evolving into reputation-as-service applications like epinion[2], but still mostly have a single-provider model. This

---

[1] http://www.ebay.com/
[2] http://www.epinion.com

development suggests that reputation systems may (and should) move next to the model of user-controlled reputation usage in various applications.

Making reputation management privacy-respecting will hopefully be the next step. We present two possibilities we tested within PRIME and that will be further elaborated and tested within PrimeLife[3].

19.2 Model of Reputation Systems

In this section we give an overview which aspects reputation systems have to cover and which design options for their implementation remain.

19.2.1 Reputation

Reputation can be assigned to various **reputation objects**. Reputation objects can have different reputation natures like individual reputation for persons, group reputation for groups of persons, product reputation for products or services etc. [SS05]. Especially one can distinguish between

- **dynamic reputation objects:** The reputation of the reputation object usually changes over time depending on how the reputation object changes. Humans belong to this class. Their reputation as sellers or buyers is, for example, collected by eBay.
- **static reputation objects:** While the reputation of the reputation object might change, the reputation object itself does not change. After such a reputation object has collected a large amount of reputation, this reputation will usually converge to a certain reputation value. Examples are non-changing products like books (e.g., collected by epinion).

Reputation can be assigned to reputation objects in two possible ways:

- **Implicit reputation** is linked to the name of the reputation object. Simply by using this name, the reputation object reaches a recognition effect by others who associate this name with a certain reputation or expectation. But this reputation is not formalised in the form of a reputation value. Examples are famous product brands.
- **Explicit Reputation** is built by explicit ratings from raters who try to subsume their experiences with the reputation object in a rating. With the help of a reputation system the ratings are aggregated to the object's reputation.

The ratings given by raters can be:

- **subjective ratings** that are influenced by the raters' subjective estimation of the reputation object.

---

[3] Privacy and Identity Management in Europe for Life (http://www.primelife.eu/), funded by the European Union in the 7th Framework Program starting March 2008.

- **objective ratings** that can be verified by all other entities than the rater at some point in time and that would have come to the same ratings.

An example for the first type of ratings is eBay while examples for the second type can be found in P2P systems, e.g. GNUnet[4] where the reply to a query leads to a positive reputation, and a reply can be proved or verified at least at the time it is sent.

Especially if the raters are humans, subjective ratings will be given.

19.2.2 Reputation Network

Let $E_1, E_2, \ldots$ be entities interacting with each other within a social network, the so-called reputation network. These entities use pseudonyms in interactions with each other. Let these pseudonyms at time t be $p_{t,1}, p_{t,2}, \ldots$. On the one hand entities within the reputation network can learn possible interaction partners' reputation from the former interaction partners of those or other entities within the network who observed the possible interaction partner. In social sciences this is called the **learning mechanism** of the reputation network [BR01]. On the other hand entities within the reputation network may control others in the reputation network by spreading information about the entities' former interactions. In social sciences this is called the **control mechanism** of the reputation network [BR01].

Both entities and interactions within the reputation network can be reputation objects. Entities and non-completed interactions are dynamic reputation objects while completed interactions are static reputation objects. Reputation systems assist reputation networks technically. We assume that they collect explicit reputation only about members who agreed on collecting it because according to [Byg02], opinions about a natural person can be seen as personal data the respective person's right on informational self-determination should be applied to. For this reason a reputation system has to assist explicit membership actions regarding a reputation network resp. system. A person must be able to apply for membership under a certain pseudonym in a reputation network and also must be able to terminate his membership.

To implement both learning and control mechanism of the reputation network, a reputation system has to offer the following actions to the members:

- **Learning mechanism through evaluation of reputation:** All members that influence the reputation of an object by their ratings, additional trusted third parties, the reputation object itself and possible future interaction partners might evaluate a reputation's object following specific rules that are fixed by the designer of the reputation system. Every evaluator might receive a different reputation of the reputation object.

---

[4] www.gnunet.org

The selection of ratings used for the evaluation of reputation depends on both the information flow of ratings in the reputation network and the trust structure on the reputation network how evaluators trust in ratings from other members. Trust in the rater is needed that he gave a correct rating according to his view on an interaction and how his subjective view and therefore subjective rating fits with the evaluator's views on interactions.

- **Control mechanism through rating:** There are two types of members who can make use of the control mechanism, the interaction partner in the form of interaction-derived reputation and possible observers in form of observed reputation [Mui03]). The system provides authorised raters with a rating function that allows them to map reputation objects to ratings. From the received ratings, the reputation of the reputation object is updated by the reputation system.

After creation of reputation it has to be stored somewhere. Reputation might be stored

- centralised at reputation servers designated for this purpose,
- locally at the device of the user whose pseudonym $p_{t,i}$ received reputation $rep(t, p_{t,i})$, or
- distributed at the devices of other users.

eBay is the typical example for central servers while GNUnet is an example for distributed storage.

The reputation selection for evaluation can be:

- **global:** This means the information flow within the reputation network is complete and every evaluator gets the same reputation of a reputation object.
- **individual:** This means an evaluator only gets a partial view on the reputation available.

In [Vos04] a simpler categorisation in four classes is made that merges the aspects of storage and data flow but we found it advisable to separate these aspects.

The rating and update of reputation has to follow specific rules fixed by the system designer. These rules usually depend on the application scenario and have to fulfil sociological and economic requirements. We abstract here from the concrete functions to allow a universal design interoperable with PRIME and various application scenarios. An overview of possible functions is for example given in [Mui03]. For an economic introduction we refer to [Del03].

This model of a reputation system interoperable with an interaction system (e.g., a community system) and an identity management system like PRIME is illustrated in Figure 19.1.

When a reputation system interoperates with an identity management system like PRIME, it is possible and intended that entities have several
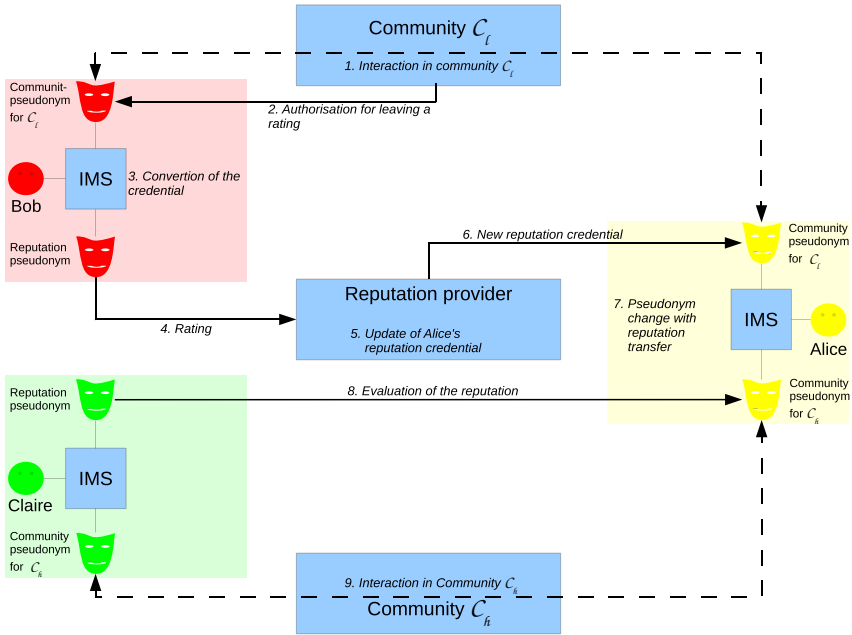
Fig. 19.1 System design

partial identities (pIDs) which cannot be linked, neither by other entities using the systems nor by the underlying system (as long as the entity does not permit this).

If there would exist only one reputation per entity, all pIDs of this entity would have the same reputation. This would ease the linking of the pIDs of one entity because of the same reputation value. Thus, having separated reputations per pID and not only one per entity is a fundamental condition for a reputation system in the context of identity management.

The use of pIDs raises the problem that a malicious entity may rate himself a lot of times using a new self-created pID for every rating in order to improve his own reputation. This kind of attack is also known as Sybil attack [Dou02]. If the reputation system is not defined carefully, it would be easy for such an attacker to improve the own reputation unwarranted. This can be limited/prevented by entrance fees or the use of once-in-a-lifetime credentials as suggested in [FR99]. We implement the latter by the identity provider from PRIME issuing such credentials. Alternatively or additionally he could also collect fees.

In the following two sections we present two implementations of the model, one with a combination of a reputation and an interaction system as one system (section 19.3) and the other with the reputation system implemented

as a stand-alone service interoperable with future identity management (section 19.4).

19.3 Reputation within BluES'n

Within PRIME, the collaborative and privacy-aware eLearning environment BluES'n (see CeL chapter) was developed. We developed a reputation system already integrated in BluES'n.

19.3.1 Characteristics of a Reputation System in the Context of Collaborative eLearning

In eLearning scenarios similar to traditional learning, a learner can trust in his teacher and materials provided by the teacher. In contrast to that in collaborative eLearning, where users interact with each other having equal rights and create common knowledge, such an assumption cannot be seen as a fact. So, if somebody wants to learn something about a topic he is not able to assess which information is correct and whom to trust. The collaborative character of BluES'n enables the participants to change their roles in the eLearning system as often as needed or desired. This way, it is possible that one and the same person may act, e.g., as tutor, learner, and author in the system. However, the users have no possibilities to check if the interaction partner has the competence for the according role before interacting with him. The same holds for learning content created by collaborative work. Because of the absence of an overall quality assurance, it cannot be assured that this content corresponds to the particular needs of a respective learner. Especially, the learners even cannot be sure if the content is correct at all. When content possesses a significant reputation value it is easier to decide if it is trustworthy or not. Reputations of users are also required as an indicator of the trustworthiness of potential interaction partners.

In eLearning, the effects of reputation are not only the learning mechanism for the evaluator and the control mechanism for the rater as outlined in section 19.2.2. But there is also an effect on the rated entity as reputation object itself: Knowing the own reputation helps an entity to figure out his own standing and can be used for self-assessment. The learning mechanism does not only help for direct trust building, but also implies group decision making, for example, who should become team leader. Group decision making needs the reputation to be global to make the decision transparent and comprehensible.

19.3.2 Basic Design of the Reputation System

With the BluES'n reputation module any content within the system can be rated. From the ratings received, the reputation of the content can be calculated by a reputation server. Since any content in BluES'n belongs to at

least one user who is the author of the content in an additional step, the user's reputation can be calculated/updated. So both content and authors are reputation objects.

The dialogue, used for the rating process, consists of a slider containing an ordered set of possible ratings. Additionally there is the possibility to give a comment together with the rating. This shall help other users to understand why the concrete rating has been given.

We allow only positive values for a single rating for several reasons. The first reason is that the usage of negative ratings may have the effect of stoning: In an eBay study the probability to receive a negative rating could be more than six times higher after having received a first negative rating. The second reason is that in an own examination we found out that the participants would favour to give positive ratings for good content than negative ratings for bad content [Jus06]. Allowing only positive ratings needs the reputation calculation to be designed in a way that the reputation is scaled in a way that there is a maximum score. Additionally to help authors of bad content to improve their content we allow for zero ratings. This kind of special rating has no influence on the overall reputation but it has to be accompanied by a comment helping the author.
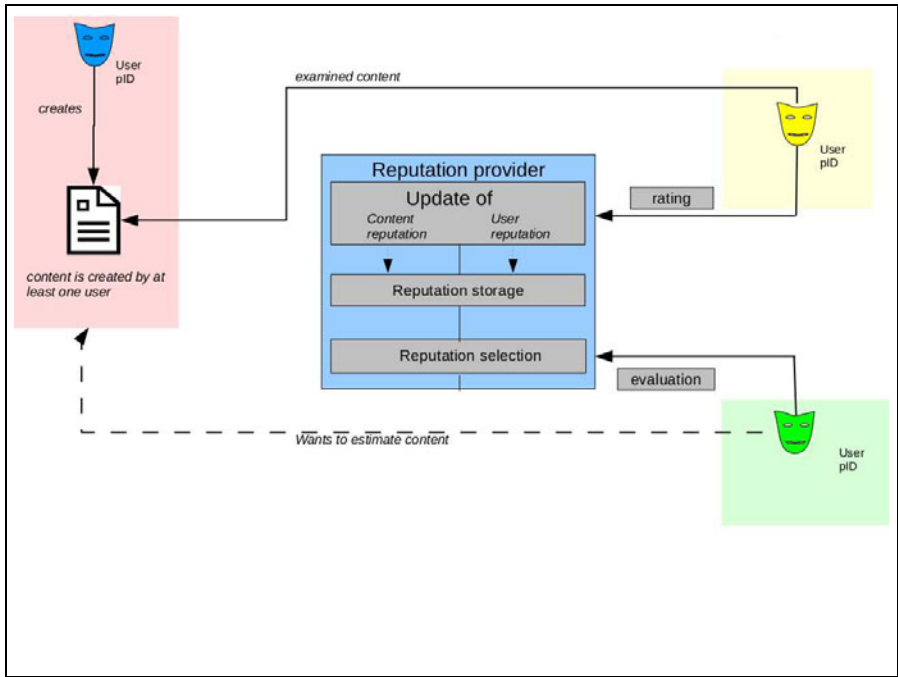
Fig. 19.2 Blues'n Design

To prevent users from making ratings in a high frequency in order to manipulate the reputation of content and thus of authors in the form of ballot stuffing we introduce an additional time factor for each relationship between rater and reputation object. That means, when a reputation object is rated, also a time-stamp is saved. This time-stamp can be compared to former timestamps of the same rater-entity combination. A repeated rating will have nearly no influence on the respective reputation object's reputation, if the period of time between the timestamps is too short.

The reputation of an author $p_{t,i}$'s content $cont(p_{t,i})$ is calculated as a function influenced by the rating provided by a user, a factor which refers to the time-stamps of the rater-entity relationship and a reducing factor which is influenced by the relation of the reputation value and the maximum score.

Based on the reputations and ratings for content also the reputation of the users (the authors) can be calculated. The only difference between the calculation of user reputation and content reputation is that user reputation takes into account all ratings given to all contents created by this user.

Figure 19.2 shows the designed system.

One possibility to avoid frauds like the Sybil attack is to factor the reputation of the rater into the calculation. That means each rating is weighted in relation to the reputation of the rater. This way, a rating with a newly-created pID, which has the minimum reputation, has almost no influence on the reputation of the to-be-rated user and thus, the negative effects can be minimised.

19.4 Reputation as Service for PRIME Applications

19.4.1 Necessary Infrastructure

The reputation service for PRIME applications [PS08] is divided into three parts: a community server as interaction system, PRIME as identity management system and a reputation provider, which is responsible for the reputation functions.

The community server allows members to interact with each other. The interaction data are stored on a central server. The framework phpBB[5] could be used as a community server for some types of interactions like discussions or the electronic form of a garage sale.

The identity management system assists the user's decision which pseudonym to use in which interaction and with which interaction partner. For using this service a registration step is necessary. After checking the user's data, the identity manager will issue a basic credential.

The reputation system can cover several communities and is able to act independently from the communities. Before interacting with each other, the interaction partners can inform themselves about the other's reputation.

---

[5] http://www.phpBB.com

Based on the basic credential, the reputation system creates a reputation credential and sends it to the user. The credential contains as attributes a pseudonym, the initial reputation, a number of free spaces for recent ratings and an expiration date. The credential is a pseudonymous convertible credential [Cha86] the user can convert to another pseudonym within the reputation network whenever he likes.

After the conversion of the reputation credential to a community pseudonym, the user can register this credential within a chosen community by showing the converted credential. The community server issues a community credential to him and he becomes a member of this community.

The communication is secured by encryption and authentication measures to ensure the confidentiality and integrity of the transmitted data.

19.4.2   System Design

The following paragraphs give an overview over the different functions of the reputation system, which are also illustrated in figure 19.3.
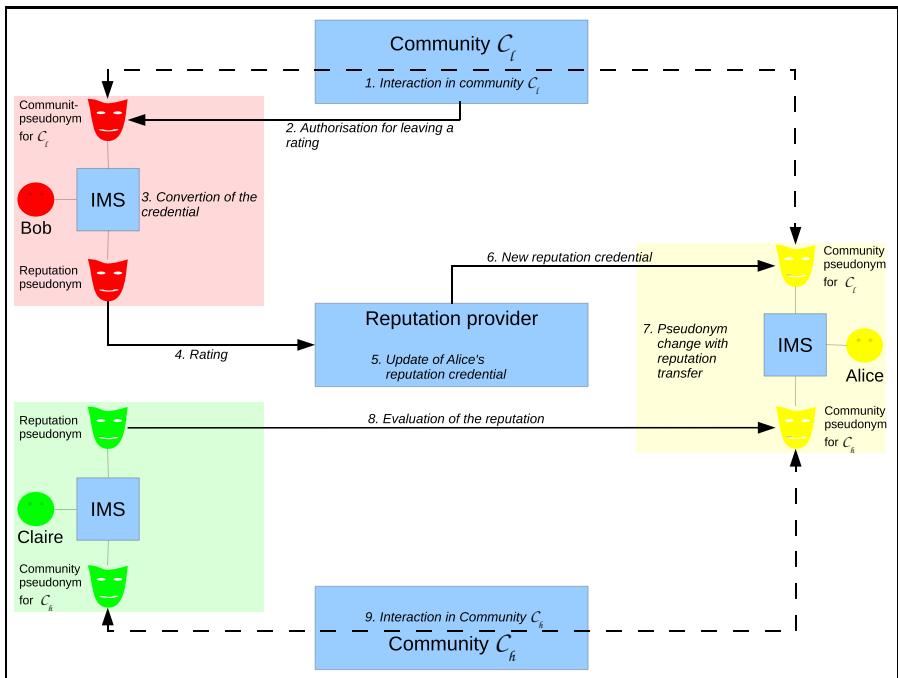
Fig. 19.3 Application-independent system design

Rating

For doing a rating, an authorisation is needed. If an interaction has taken place (Fig. 19.3, step 1) and was finished, this authorisation is given to the participants by the community server (Fig. 19.3, step 2). To ensure the unlinkability of the interaction and reputation data, convertible credentials are used. The credential states the pseudonym for which a rating can be left.

This interaction credential can be converted from the community pseudonym to the reputation pseudonym of the corresponding user to guarantee unlinkability (Fig. 19.3, step 3). For the actual rating, this credential, the rating and the reputation object's pseudonym are sent to the reputation system where the data are stored until the reputation object is updating his reputation (Fig. 19.3, step 4).

Update Reputation

The reputation credential has to be updated after a fixed number of ratings $k \geq 1$ has been given to that credential (Fig. 19.3, step 5). According to [Del06] it make sense economically not to update the reputation after every rating but after $k > 1$ ratings. Besides this, it increases the unlinkability of the reputation object.

To perform the update, the reputation credential has to be sent to the reputation system. The initiation might start by the member or by the reputation provider. Inside the credential the attribute containing the reputation has to be updated and the new rating has to be added as attribute resp. substitute one of the existing expired rating attributes. The reputation provider does not need to know the content of the reputation credential. Only the relationship between the old and the new credential must be guaranteed. Therefore the calculation is possible on encrypted values, if the reputation algorithm is homomorphic regarding the encryption.

The reputation computation algorithm can be chosen arbitrarily by paying attention to the fact that users are recognisable by their reputation, even if they use convertible credentials to reach unlinkability of their actions. For this reason the reputation and rating set have to be small enough to reach sufficiently large anonymity sets. Details about this idea are outlined in [Ste06].

After updating the credential, the provider sends the new reputation credential to the reputation object (Fig. 19.3, step 6). The old reputation credential would still be valid if it did not contain the attribute for the expiration date.

Pseudonym Change

To increase the unlinkability between different interactions of a user, the change of the credential pseudonym should be possible. To ensure the significance of the reputation system, the aggregated reputation has to be transferred as suggested in [Ste06] (fig. 19.3, step 7).

The transfer is realised by convertible credentials. This realisation allows the user to convert the credential to a new pseudonym without trusting the reputation provider. This pseudonym change only makes sense if a large number of members with the same attributes (e.g. the reputation) change their pseudonyms at the same time to guarantee an appropriate anonymity set. For this reason the sets of possible ratings and reputation values are limited.

If a member wants to change his pseudonym when a rating has been left for him at the reputation provider, it could not be guaranteed that the mapping between the new pseudonym and the rating could be made. Therefore the reputation provider has to authorise the pseudonym change.

Reputation Evaluation

Before deciding on an interaction, the reputation of the possible interaction partner could be evaluated pseudonymously, after the holder has sent his reputation credential to the evaluator (Fig. 19.3, step 8).

To augment the availability of the reputation, a supplementary storing at the reputation server or the community server should be possible. But this needs the user to appoint authorisation to other members of the community to see his reputation.

Leaving the Reputation System

Every member can always leave the community or reputation network. But if a member has a reputation less then the initial one at this point, the identity should be banned by the identity provider, so that this member could either not get a new basic pseudonym to register with a reputation network or with a community, or if he gets one, this pseudonym will get the old reputation.

19.5   Outlook

The basis and preconditions to design reputation systems compliant to privacy-enhancing user-controlled identity management were introduced in this chapter. This concept becomes more and more important with the growing number of applications which need reputation systems.

Our future research, especially within PrimeLife, will concentrate on the interoperability between reputation systems and identity management to allow an easier and more privacy-respecting handling of users' various identities and reputations.

Regarding the system architecture we will try to develop distributed alternatives to the central reputation providers. This will hopefully allow for individual reputation additionally to the global reputation in our current system designs.

Human-Computer Interaction

Simone Fischer-Hübner[1], John Sören Pettersson[1], Mike Bergmann[2],
Marit Hansen[3], Siani Pearson[4], and Marco Casassa Mont[4]

[1] Karlstad University
[2] TU Dresden
[3] ULD
[4] HP Labs

20.1 Introduction

An important critical success factor for PRIME technology will be user-friendly and intelligible user interfaces that convey and enhance trust. Such user interfaces have to meet challenges such as:

- **User-friendly representation of complex PET concepts:** PRIME and other privacy-enhancing technologies (PETs) are based on technical concepts or constructs such as pseudonyms, unlinkability, anonymous credentials as well as policy negotiation and management that are unfamiliar to many end users and often do not fit their mental pictures of what is technically feasible. Informational self-determination means that users are able to decide how their personal data are used. This should not necessarily have to involve determining how technicalities such as pseudonymisation are carried out. From a usability perspective, such technicalities should on the contrary rather be invisible to the users. However, when it comes to understanding the risk of being identified across different interactions with one or several service providers, some sort of notion about digital identity must be understood by the user.
- **Provision of security:** The PRIME user interfaces also need to be "secure" in the sense that they should have reasonable countermeasures against common types of Internet fraud attacks, such as phishing and spoofing.

- **Mapping legal requirements:** Another important task of the user interfaces is to enforce and promote legal principles such as informed consent or transparency, so that the user interfaces are not only privacy-compliant, but also enhance the users' understanding, awareness and control. For enforcing the privacy principle of transparency as a prerequisite for user control, a special challenge is to design user interfaces that are informative while user-friendly: Users must be well informed about the consequences when releasing data, and consequently there are legal requirements for providing information to the users (e.g., Art. 10, 11 EU Directive 95/46/EC [Cou95]) that need to be met by the PRIME user interfaces. Nevertheless, users should not be confronted with excessive or badly structured information that is usually perceived as bothersome and ignored by the users.
- **Mediation of Trust:** Usability tests of early PRIME prototypes have shown that there are problems to make people trust the claims about the privacy enhancing features of the systems (see D6.1.b, Pettersson et al. [PFHD+05]). Similar findings of a lack of trust were also recently reported by Günther et al. [GS05] in a study on the perception of user control with privacy-enhancing identity management solutions for RFID environments, even though the test users considered the PETs in this study fairly easy to use.

In this chapter, we will present how the HCI research in PRIME has addressed some of those challenges and what our main research contributions in those areas have been. Besides, we will also discuss some open HCI research issues for privacy-enhancing identity management that have not sufficiently been solved in PRIME yet.

20.2   Related Work

Scanning the fields of privacy and HCI provides some interesting articles illuminating intriguing intersections. In general, the works reported have had different focuses, but in the last few years also some comprehensive sources have appeared, such as Microsoft's Privacy Guidelines for Developing Software Products and Services [Inc06] and the SOUPS conference, Symposium on Usable Privacy and Software, held annually since 2005 at Carnegie Mellon University in Pittsburgh. One may also mention Iachello's and Hong's survey of HCI research on end-user privacy [IH07]. Below, papers of various origins are presented under the headlines of the four challenges listed above.

Identity management components sometimes figure prominently in papers on the usability of security systems and are therefore of relevance to PRIME. Some other works on security and HCI are also mentioned here.

20.2.1   User-Friendly Representation of Policy Management with the Help of Default Settings

Several authors have noticed the difficulties for end-users to set security parameters while heavily simplified setting functions do not provide an adequate set of security levels (Kröger, 1999; Whitten & Tygar [WT99], Jendricke & Gerd tom Markotten [JtM00], Gerd tom Markotten [Ger02], several works by Steven Furnell, i.a. [Fur04a, Fur04b, Fur05, FJK06], Cranor & Garfinkel [CG05]). In Nielsen's report "User Education Is Not the Answer to Security Problems" he states that accountability of security cannot be the users' responsibility [Nie04]. He adheres to common recommendations about making security a built-in feature of all computing elements and turning on all security settings by default "since most people don't mess with defaults". Then, make it easy to modify settings so that users can get trusted things done without having to open a "wide hole for everybody". This sounds indeed as the working premises taken by PRIME with relationship 'pseudonyms' playing an important role in the user interface proposals while the default is total anonymity. The P3P privacy bird (www.privacybird.org) provides some predefined P3P preference settings, which can be customised by the user during the installation process and via the privacy bird menu. However, in contrast to the approach that we have taken, P3P does not permit to define more fine-grained privacy preferences that could for instance be conditioned on individual data controllers and data values. The privacy bird also does not allow to change privacy preference settings semi-automatically "on the fly". Hence, it is not surprising that a Privacy Bird User Study reported that while those users who changed their privacy settings reported it was relatively easy to do so, only a minority reported changing them several times [CAG02].

20.2.2   Secure Interfaces

It has been noted by many that a substantial proportion of the population is afraid of using e-services because they fear fraudulence and privacy crimes. Looking at the literature that started to appear around mid 1990 on e-shoppers' behaviour, the marketing perspective has received much attention for a long time, while, however, research on real problems for users/customers was not prominent. Making a web site sell by extensive usability analysis and by making it mediate a sense of trustworthiness were key aspects. This was research that informed the service providers. Little was done on how to inform users of when to trust web sites – there are indeed problems of mediating trustworthy information because the user must understand which indicator to look at. Unscrupulous web site owners naturally will use all means available to fool innocent visitors. For instance, they will use trust signs they have not been awarded or simply copy the appearance of other sites.

In an illuminating study on this topic in which 22 participants were shown 20 web sites and asked to determine which ones were fraudulent, the researchers "found that 23% of the participants did not look at browser-based

cues such as the address bar, status bar and the security indicators, leading to incorrect choices 40% of the time. We also found that some visual deception attacks can fool even the most sophisticated users." [DTH06] Another study concludes: "We confirm prior findings that users ignore HTTPS indicators: no participants withheld their passwords when these indicators were removed. We present the first empirical investigation of site-authentication images, and we find them to be ineffective: even when we removed them, 92% participants who used their own accounts entered their passwords." [SDOF07]

Thus, the concern for users turn out to be a question of how to provide 'secure interfaces' where trustworthy information can be provided. Wu, Miller, and Little [WML06] let test users enter sensitive information online via a browser sidebar. In a usability study, this solution "decreased the spoof rate of typical phishing attacks from 63% to 7%." However, spoofing the 'secure' sidebar itself turned out to be an effective attack. There has been some further discussion about how well security and privacy indicators work; see the overview by Cranor [Cra06]. As suggested by Djamija and Dusseualt ([DD08], identity management systems should support mutual authentication rather than only focusing on user authentication. This means that also the services sides need to authenticate themselves for the users. In section 20.4 we describe the approaches elaborated within PRIME.

20.2.3   Mapping Legal Privacy Requirements

In the PISA project ("Privacy Incorporated Software Agent", an EU FP5 project), it has been studied in detail how privacy principles derived from the EU Data Protection Directive 95/46/EC can be translated into HCI requirements and what are possible design solutions to meet those requirements [PK03]. The derived HCI requirements were grouped into the four categories of comprehension (to understand, or know), consciousness (be aware or informed), control (to manipulate, or be empowered) and consent (to agree).

In the PRIME project, we have used and extended these privacy principles and HCI requirements from the PISA project to derive proposed UI design solutions for PRIME (see [WP608]). The PISA project investigated in particular also user agreements for obtaining informed user consent and introduced the concept of 'Just-In-Time-Click-Through Agreements' (JITC-TAs). "The main feature of a JITCTA is not to provide a large, complete list of service terms but instead to confirm the understanding or consent on an as-needed basis. These small agreements are easier for the user to read and process, and facilitate a better understanding of the decision being made in-context" [PK03, PKHvB02]. The concept of a JITCTA was also used for the PRIME HCI proposals using the "PRIME Send Personal Data?" dialogue boxes (see [WP608]), which will be discussed in section 20.5.1.1.

The Article 29 Data Protection Working Party has also investigated what information should be provided in what form to users in order to fulfill all legal provisions of the EU Data Protection Directive 95/46/EC for ensuring

that individuals are informed of their rights to data protection [Art04]. The Art. 29 Working Party recommends providing information in a "multi-layered format under which each layer should offer individuals the information needed to understand their position and make decisions". They suggest three layers of information provided to individuals: The short notice (layer 1) must offer individuals the core information required under Article 10 of the Directive 95/46/EC, which includes at least the identity of the controller and the purpose of processing. In addition, a clear indication must be given as to how the individual can access additional information. The condensed notice (layer 2) includes in addition all other relevant information required by Art. 10 of the Directive such as the recipients or categories of recipients, whether replies to questions are obligatory or voluntary and information about the individual's rights. The full notice (layer 3) includes in addition to layers 1 and 2 also "national legal requirements and specificities." It could be noted that the so-called 'condensed notice' contains a 'full' set of information as judged by the EU Directive, and it has been used as the 'full' version in PRIME sketches.

The Art. 29 Working Party sees short privacy notices as legally acceptable within a multi-layered structure that, in its totality, offers compliance. JITCTAs as defined in the PISA project are in fact corresponding to such short privacy notices. Within PRIME, we have followed the Working Party's recommendations to use multi-layered privacy notices in its design proposals (see below).

20.2.4   Mediation of Trust

Recommendations from third parties have in some studies been found to be a trust-giving factor [Tur03]. It might be possible to manipulate customer ratings, but there are organisations issuing trust marks of various sorts; of particular relevance here are of course 'privacy seals'. Currently there are different standards for such seals. The EU project EuroPriSe[1] aims at establishing a common European standard for the process by which service providers earn their seals). In section 20.6 we discuss how relying on automated checks of assurances given by service providers could enable users to evaluate if they dare to use a certain service.

20.3   Challenge I: User-Friendly Representation of Complex PET Concepts

In this section, we will concentrate on user-friendly management of privacy preferences. First, we will present a set of predefined privacy preferences, from which users can choose and then discuss UI approaches that allow one to customise privacy preferences semi-automatically "on the fly" when a services

---

[1] www.european-privacy-seal.eu

side has been contacted and is requesting personal data. Those predefined privacy preferences should represent the privacy interests that users might have for various applications using basic settings for managing most of the identity management tasks and should include the most privacy-friendly options. Then, we will present alternative UI-Paradigms for presenting such privacy preferences to the end users in order to simplify for them the process of choosing the right privacy preferences fitting their demands when contacting a services side.

20.3.1   Simplified Policy Handling

Even if users' real IP addresses are hidden through the use of anonymisation services and if they use pseudonyms when contacting web sites, they might have to disclose personal attributes constituting a partial identity at some services sides.

In PRIME, the user's release policy (or his/her so-called "privacy preferences") defines the user's preferences regarding the release of his/her data. At the services side, a so-called data handling policy (or simply "privacy policy") specifies how and what data are used by the services side. If personal data are requested from a user by a services side, the PRIME user-side system can compare ("match") the services side's privacy policy with the user's release policy (privacy preferences) and warn the user in case of a mismatch. For ordinary users defining and adapting a privacy-friendly release policy is a complex and error-prone task which usually requires some expertise about basic legal privacy concepts and principles. In the non-electronic world no equivalent task exists, which means that ordinary users have usually no experiences with the definition and management of their release policies. Without assistance, most users would not define and use release policies at all or could accidentally define or choose a release policy, which is not as privacy friendly as they would like it to be.

Therefore we have derived a set of four predefined "standard" privacy preferences from which a user can choose and which he/she can fill in with concrete data values or which he/she could customise "on the fly" and store under a new name. The predefined privacy preferences (so-called "PrivPrefs") define what types of data may be released for what specific purposes under what specific conditions. In addition to those settings which will be compared with the privacy policies of services sides when they request personal data from the user, our privacy preferences also set the type of pseudonymity/level of linkability to be used. Our set of predefined privacy preferences should represent the users' privacy interests and also includes the most privacy-friendly options for acting anonymously or for releasing as little information as needed for a certain service.

More precisely, the following PrivPrefs have been defined: The first one is called "PRIME-Anonymous" which should be activated by default if no other PrivPref has been chosen by the user. It is useful, for example, for anonymous

browsing. With the PrivPref PRIME-Anonymous, no personally identifiable data is actively released by default. Transaction pseudonyms are used, i.e. user actions are not linkable beyond the transaction.

Another PrivPref is "PRIME Returning Visitor", which is useful if the user does not want to directly release personally identifiable data, but would like to allow services sides to store settings, which can then be utilised for later visits. With the PrivPref Returning Visitor, no personal data that are directly identifying the user are released. What might however be released are for instance data about personal settings. Besides, visits to the same web site are linkable through the use of role-relationship pseudonyms.

The first two PrivPrefs were designed for applications, where personal data are not directly requested from the user. However, for many e-applications such as e-shopping, e-health or e-government applications, users usually have to provide personal data. For such applications the most privacy-friendly data release policy will be one which reveals only the minimal amount of data needed for providing the requested services, where the data will only be retained until the services are completed and will not be forwarded to other third parties. Besides, different transaction pseudonyms should be used for different transactions.

The question of what the minimal amount of data is, varies between different applications/services and is dependent on the purposes for which the applications will need to collect and process personal data. Hence, we have to define specific PrivPrefs for specific applications. As an example, we defined the third PrivPref called "PRIME Minimal Shopping" for an e-shopping service, where the customer would like to release only the minimal amount of data needed for this service, which should be retained only until the service is completed. For providing an e-shopping service, different personal data items will be needed for the purposes of the sub-tasks Registering/Placing an order, Delivery (physical or electronic) and Payment. Today, e-shopping sites are usually not only collecting data about the placed orders from the customers, but are also requesting payment data and address data from their customers, which they then forward to the payment providers and delivery services which are cooperating with them. However, there is usually no need for the e-shop vendor to know the customer's address or payment information. In [Ber08], we describe a more privacy-enhanced solution, in which the e-shop requests only the data needed for placing the order, whereas payment details (e.g., credit card details) are requested directly by the payment provider and address details are directly requested by the delivery service. This means that in such a solution personal data are not forwarded to other third parties, but are instead directly requested by the parties that need to process these data.

Table 20.1 lists the data types that are typically needed for the purposes of the e-shopping sub tasks. Our PRIME Minimal Shopping PrivPref only allows data collection for the purposes Order Registration, Delivery (physical or electronic) and Payment, and restricts the type of data to be collected to those listed for those purposes in Table 20.1 (which are assumed to be the

Fig. 20.1 An example "PRIME Send Personal Data" assistant

minimal amount of data needed for those purposes). If more types of data are requested for those or other purposes (as stated in the services sides' policies), and if the user has chosen the PRIME Minimal Shopping PrivPref he/she will be warned that more data are requested than needed. In addition, the PRIME Minimal Shopping PrivPref includes the preference setting that personal data should not be forwarded to other third parties, which means that the users will also be warned if the services side's privacy policy allows for such data transfers.

Finally, we have also predefined a PrivPref called "PRIME Profiled Shopping", which could be chosen by users who agree to release more data than needed for the primary e-shopping service, usually in return to other benefits, such as bonus points. With this PrivPref, the user would also agree to release his/her address details for the purpose "Marketing" and would also agree that the Shop could process information about his/her orders for the purpose "Profiling" as specified in Table 20.1.

As mentioned above, the PrivPref PRIME-Anonymous is activated by default, if no other PrivPref has been chosen by the user. The user should have the possibilities to select another PrivPref before or when contacting a services side (for this case we will discuss UI approaches in the next section) or after having contacted a site. The predefined PrivPrefs PRIME Minimal Shopping and PRIME Profiled Shopping are from the start only defining what data types (rather than concrete data values) may be released for what purposes. For a simplified handling of PrivPrefs, it should however be possible to customise the PrivPrefs and fill in concrete data values "on the fly" rather than demanding that the user has to fill in the values by hand before he/she can use those PrivPrefs. This means that when a services side is requesting personal data and the user fills in data values in a form, such as the "PRIME Send Personal Data?" dialogue form (see Figure 20.1), he/she will be asked whether he/she would like to save these data values in the PrivPref that is currently activated. In order to guide the user through these phases of PrivPref

selection, data collection by usually different parties such as an e-shop, payment provider and delivery service and of PrivPref customisation, a wizard-based user interface approach, has been developed. A short pilot user test confirmed that users percept the splitting into subtasks as simplifying the process and increasing the transparency. Further user tests should examine this statement in more detail.

As shown in Figure 20.1 the wizard informs the user about the overall procedure. It collects all the required personal data by the different parties, shows the dedicated purposes of the data requests and allows to walk through the different stages to check the settings made before. It optionally shows also available information about the service provider (e.g. seals or reputation data), about the requested certificates as well as the full data handling policies and obligations. In our example the dialogue contains sections presenting contact information about data recipient, stated purposes and required personal data. The wizard in Figure 20.1 handles the dedicated data request made by the service provider.

In comparison to the management of privacy preferences by the P3P privacy bird or other P3P user agents, our PrivPref approach has particularly the following advantages:

- Different PrivPrefs can be defined for different services sides, whereas P3P only allows the user to define one privacy preference setting which then applies for all web sites that he/she visits. Besides, the PrivPrefs allow defining also preference settings on the granularity of concrete data values. Hence, the user can define more fine-grained privacy preferences which provides him/her better privacy protection;
- PrivPrefs can be changed and filled in with data values semi-automatically "on the fly", which simplifies the process of changing and customising privacy preferences;
- The predefined PrivPrefs allow one to check whether a services side's privacy policy is conform with the privacy principle of data minimisation and inform users if more data is requested than needed;
- A PrivPref allows also for setting preferences concerning the linkability of the transaction pseudonyms to be used when this PrivPref is activated.

20.3.2 UI Paradigms for Presenting Privacy Preferences

After we have in the last section focused on the content and customisation of predefined privacy policies, we will in this section proceed with UI Paradigms for presenting these privacy preferences to the end users, in order to make the choice of appropriate privacy preferences more intuitive for them (see also [PFHD+05, FHPB+07]).

In PRIME, we have bundled preference settings for personal data and pseudonym types as so-called roles or areas in three main UI paradigms, namely the *role-based*, the *bookmark-based* and the *townmap-based* paradigms.

Table 20.1 Data types (tentatively suggested) in relation to stated purposes

Purpose	Data types	Comments
Order Registration	Ordered items	For shopping cart
	Session pseudonyms	similar to a session cookie
Physical Delivery	Name	Alternative 1
	Address (full)	
	Pin code (received from service prov.)	Alternative 2
	Pick up point	
Electronic Delivery	Email address	Alternative 1
	Internet link (user is given a link)	Alternative 2
Payment	Credit card info	The alternatives here are not mutually exclusive
	Bank account info	
	(anonymous) e-coins	
	Bonus points	
Registration	User name (automatically generated)	This is the minimal need
	Password (automatically generated)	
Marketing	Email address	
	Telephone number	
	Name (for physical contact)	
	Address (in combination with name)	
Statistical	All data types except Name and full forms of personal and telephone numbers	Should be anonymous, else "Marketing"
Profiling	Ordered items	Here, the user accepts profil-
	User name / user's pseudonym	ing
	(possibly more data types)	

The first two paradigms are traditionally styled while the third one is based on the metaphor of a town map and is an attempt to make preference settings more accessible and, hopefully, understandable to users. On the other hand, the two latter ones share a common approach to the use of preference settings, namely that the selection among the different preference settings (roles and areas, respectively) is implicit when connecting to the service providers. The three paradigms are presented in the three following subsections.

20.3.2.1   Role-Based Paradigm

Role-based means that user control of data disclosure is primarily carried out via the 'roles' described above which function like identity cards that allow for pseudonymous contacts. Within a role, the user can set and utilise

different disclosure preferences for different data types. The user then has to select the role he will be acting under when contacting service providers, and whenever he thinks that this role is inappropriate, he has to select one of his/her other roles. The UI paradigm was first embodied in an early user-side prototype called DRIM (Dresden Identity Management [CK03]) where the IDM functions were displayed in side bars of an ordinary Internet browser (Netscape).

20.3.2.2 Bookmark-Based Paradigm

Within PRIME, a bookmark-based approach for enabling easy and intuitive ways of privacy preferences selection when a user contacts a web site has been explored. In this bookmark-based UI paradigm embodied in PRIME mock-ups, the user can attach privacy preferences to ordinary bookmarks ("Favorites" in Internet Explorer). In this way there is, during ordinary web browsing, no extra step of selecting privacy preferences that should apply when visiting that site.

Figure 20.2 shows this approach. It offers the user regular access (clicking on the name of the service) as well as alternative access (clicking on the icons) with a different privacy preference settings for these bookmarked web sites. The PRIME user-side identity management system handles the appropriate responses. The masked man symbolises the predefined privacy preference "PRIME-Anonymous" with the completely anonymous interactions and a pseudonym for each visit (transaction pseudonymity). This prevents the website to link the user to his/her previous visits (unless personal data, such as user-name and password, are explicitly given during these interactions). The partly-hidden face at the IKEA bookmark invokes the predefined privacy preference setting "PRIME Returning Visitor" with the use of the previously-used pseudonym for this website (relationship pseudonymity) so that the website can see that it is a returning visitor.

Fig. 20.2 Bookmark list with icons for privacy preferences

By using the predefined privacy preference "PRIME-Anonymous" based on transactional pseudonyms as the default, the bookmark-based approach allows the privacy-enhancing functions to be switched on from start even

if the user is not prepared to actively select among them. The same approach of reusing or creating new pseudonyms can also be implemented in the browser's address field by incorporating multiple 'Go' buttons (see [WP608, PFHD$^+$05]).

Fig. 20.3 Townmap

20.3.2.3   TownMap-Based Paradigm

The bookmark solution can also be used in graphical representations, for instance in the TownMap, where different areas on the map represent different default privacy preference settings. In the TownMap of Figure 20.3 the user's 'Neighborhood' represents the area (web sites) where the user is more 'recognisable' than in 'Public' places. Predefined areas are the Neighbourhood (where relationship pseudonymity is used by default), the public area (where transactional pseudonymity is used by default), and the work area (where relationship pseudonymity is used), each with different default privacy preference settings. Individual bookmarks or lists with bookmark menus are symbolized by houses. The user also has his/her own house in the map (a prominent house at the baseline). The approach to use different default privacy preference settings for different areas within a town should make it easier for a novice to see and select the options available once he has grasped the TownMap metaphor [BRP05, PFHD$^+$05]. In 2005, preference tests were conducted at Karlstad University (with 34 test persons) and UC Irvine (with 27 test persons) using user interface animations, where groups of test participants could see identity management carried out in the traditionally-styled user interface and the also in the TownMap. While the traditionally-styled user interfaces got in general more positive responses, the test results also

revealed that young Internet users liked the idea of being able to switch between the two kinds of interfaces.

20.4 Challenge II: Secure Interfaces

Within PRIME, Camenisch et al. [CasSZ06] have conducted research on secure interfaces for credential selection. Part of their research has been the development of a contextual user interface, which provides a strong visual reference to the transaction with which it is associated. This allows a user to get engaged in several transactions at the same time with different identities without getting confused to which transaction a user interface belongs (this might for instance be a problem with pop ups: If a user opens two tabs for the same site at the same time, and both present popups, it will be unclear which popup belongs to which session [CasSZ06]). Camenisch et al. have developed user interface proposals that make such a visual link to the transaction through *context* by placing a specially-designed PRIME button directly on the web page, which the user has to click in order to start a transaction (see Figure 20.4).

The issue of possible spoofing of the contextual user interface is addressed by introducing web page independent and web page unaccessible visualization areas. This is for instance possible if the interface is implemented as a browser extension, as it can then employ methods which are unavailable to a malicious web page script, e.g. it can change the bottom toolbar or the URL toolbar when the menu has become activated. In the mockups by Camenisch et al., a blinking prime logo was added in the URL bar and the status bar. Eyetracking user tests of those mockups conducted at Karlstad University confirmed however the problem that users usually do not pay attention to such indicators placed in the toolbars. However, as Camenisch et al. point out, even if a malicious web page creates a spoofed menu, it will be unable to access the user's credentials and unable to induce the user's browser to send the user's information. The reason for this is that only the browser extension can interact with the wallet application which stores the user's credentials and performs the cryptographic protocols.

Help to detect spoofing attacks can be provided if in case the identity management system detects that this web service has never been visited by the user before, it notifies the user about this (as also Microsoft CardSpace does). If for instance a phishing site successfully redirects the user's request, e.g. from www.paypal.com (the original contact the user likes to connect to) to www.paypal-customer-care.com (a faked web presence aiming to mislead the user), the system should alert the user because of the new and unknown communication contact. The UI proposals by Camenisch et al. display such a feedback alerting the user of a suspected attempt of fraud visibly inside the content area of the user interface.
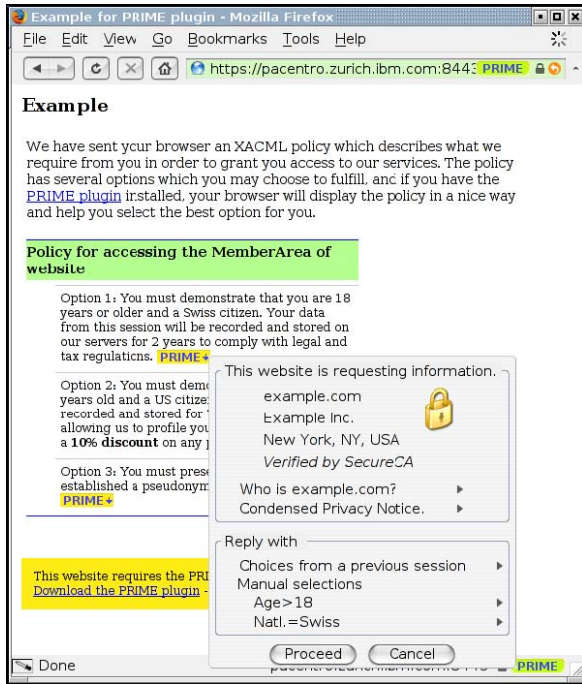
Fig. 20.4 A proposal for a secure context menu based user interface

20.5   Challenge III: Mapping Legal Privacy Requirements

As pointed out in section 20.2.3, our HCI research has built on the results of the PISA project on how to map legal privacy principles to HCI requirements and possible HCI design solutions by using and extending the privacy principles and corresponding HCI requirements and proposing corresponding PRIME UI solutions (see also Chapter 4 in [WP608]). In this section, we restrict ourselves to discussing the mapping of some important legal privacy principles to PRIME UI solutions, namely the provision for obtaining informed consent as a legitimization for data processing and the privacy principle of transparency, which encompasses the rights of the individuals to access, rectify, block and/or erase their data (see also [PFHD+05, Cou95]).

20.5.1   Obtaining Informed Consent

"Unambiguous", "explicit" or "informed" consent by the individual is often a prerequisite for the lawful data processing (see for instance Art. 7.a EU Directive 95/46/C or Art. 9 EU Directive 2002/58/EC). Informed user consent is also seen as an HCI requirement in [PK03]. Art. 10 of the EU Data

Protection Directive 95/46/EC requires that individuals from whom personal data will be collected have to be informed about the identity of the controller, the purposes of the data processing – except when individuals are already aware – and about further information in so far, as such further information is necessary, having regard to the specific circumstances in which the data are collected, to guarantee fair data processing. Web sites of data controllers within the EU have to provide privacy notices or links to privacy notices that display this information.

20.5.1.1   Informed Click-Through Agreement

JITCTAs as defined in the PISA project constitute a possible solution for obtaining consent by the user. Also two-clicks (i.e. one click to confirm that one is aware of the proposed processing, and a further one to consent to it) or ticking a box have been suggested by different European legal experts and data commissioners as a means for representing the individual's consent (see also Chapter 2 in [FHP04]).
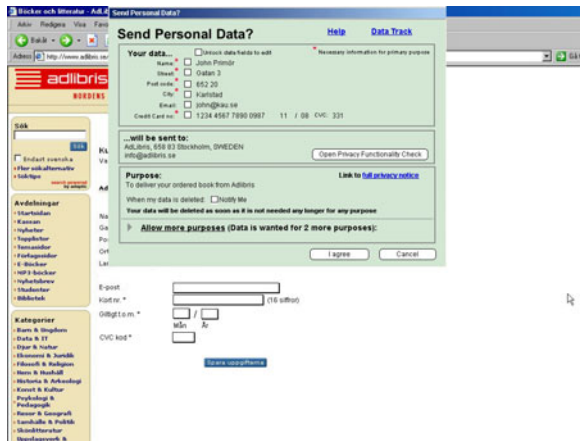
Fig. 20.5 "PRIME Send Personal Data?" dialogue

The "PRIME Send Personal Data?" window used in PRIME as illustrated in Figure 20.5 corresponds with its form and content to a JITCTA and is following the approach of multi-layered privacy notices. For the PRIME mockups and prototypes, we developed the "PRIME Send Personal Data?" window as illustrated in Figures 20.6 and 20.7, which corresponds with its form and content to a JITCTA and is following the approach of multi-layered privacy notices as suggested by the Art.29 working Party. The top layer displayed in the "PRIME Send Personal Data?" window provides all the core information

to the user required under Art. 10 of the EU Directive 95/46/EC (identity of the controller, purposes of data processing). Besides, it contains a link to the full privacy notice, which contains all information required by Art. 10 of the EU Directive 95/46/EC and other applicable laws (such as Art.4 of Directive 97/7/EC on the protection of the consumers in respect to distance contracts). Each PRIME-enabled server side should make a complete privacy policy available in computer-readable form (e.g. in XML-format), which permits that the policy display will be in a language chosen by the user. Hence, all information required by the EU Directive 95/46/EC is provided when the user is requested to agree to the data disclosure by clicking the "I agree" button. In this way, the legal requirements for an informed consent can be satisfied.

Fig. 20.6 A purpose-sensitive "PRIME Send Personal Data?" dialogue window

The PRIME solution with one uniform dialogue "PRIME Send Personal Data?" across different web sites makes it also possible to harmonise the field names and the layout of the data entry fields for all PRIME-enabled services. It interprets the data fields requested by the service provider and keeps track of what the user enters. Several 'intervening' user interfaces have been prototyped in the PRIME project to support the user in releasing data while maintaining an acceptable level of privacy. The one in Figure 20.5 is one example, while the one in Figure 20.6 represents one of the last designs developed within PRIME – a design, where it is supposed that there is a PRIME

standard list telling which data types are needed for which data processing purposes; this to make it possible for the PRIME system to notify the user if some data requests are excessive. (The scrollbar to the right is just a mock-up feature to indicate that the window may be vertically shorter than displayed in the figure.).

In Figure 20.6 the user has at some earlier point selected a predefined preference setting called "PRIME Profiled Shopping". This privacy preferences contain a list of the (few) data processing purposes, for which data of a certain type may be collected, so that the "PRIME Send Personal Data?" dialogue window can notify the user if the web site asks for data which are intended for data processing outside the scope of the preference setting. This is not the case in Figure 20.6: all three purposes are permitted by the privacy preference "PRIME Profiled Shopping", but the data type "telephone number" does not necessarily belong to the purposes of order registration and delivery, even if it is not totally unreasonably for the service provider to ask for such data for these purposes – at each instance, a "What to do" link helps the inexperienced user to decide whether she should edit the data fields or simply click the "Cancel" button.
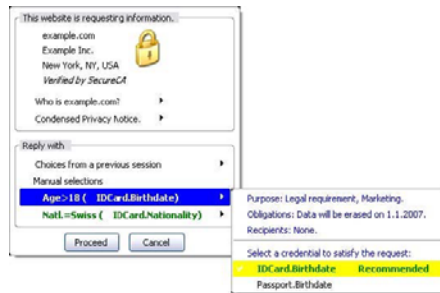
Fig. 20.7 Menu-based Approach for selecting Credentials

20.5.1.2 Consent via Menu-Based Selection

An ordinary click-through window may cause users to click the "I Accept" button too easily if the preference settings have filled in all the requested data for him/her. Putting up "Are you really sure?" boxes does not resolve the problem as people may often click the OK button even more automatically if they have to go through an extra dialogue box every time [Ras00]. Presenting data items in cascading menus to select data or credentials, as proposed by [CasSZ06] and shown in Figure 20.7 has the effect that the user must read the text for making the menu choices, which means that in this case he/she should make more conscious selections. Naturally, such cascading

context menus would then need to also include the other information that is relevant for data disclosures, and therefore the cascading context menus depicted in Figure 20.7 are also following the Art. 29 WP recommendation for a multi-layered structuring of privacy policies.

However, this user interface design is not suitable if many data fields have to be filled; the design is intended as a special feature for very simple data requests where the user might have to select among a few credentials asserting a specific data claim. (It has not been integrated with the PRIME Integrated Prototype).

20.5.1.3   Consent by Drag-and-Drop Agreements

"Drag-and-Drop Agreements" (DADAs) were also elaborated in PRIME as a method for raising the consciousness about the nature of data disclosure in conjunction with the TownMap metaphor based UI design paradigm (Figure 20.8). Symbols were used to represent personal data – this allowed users to visibly drag-and-drop data to icons representing the receivers. Here, the user not only has to pick a set of predefined data (corresponding to clicking "I Accept" or "I Agree" in a pop-up window), but choose the right personal data symbol(s) and drop them on the right recipient symbol. These explicit actions to some extent offer a guarantee for more conscious user consent.
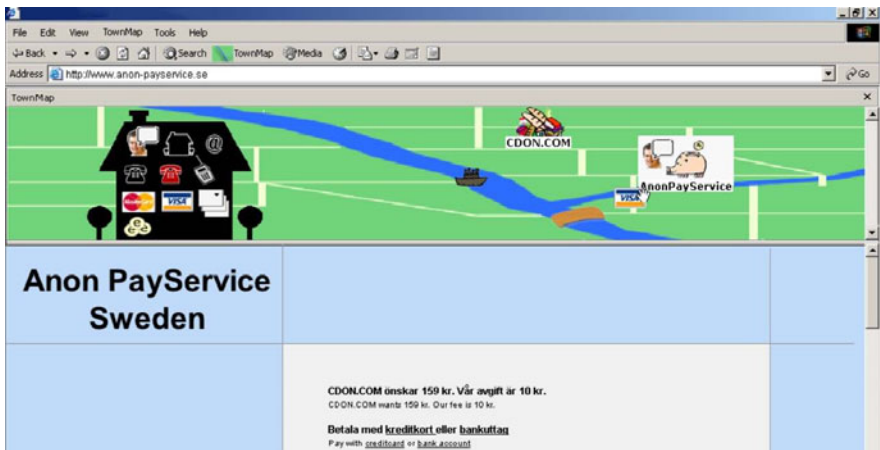
Fig. 20.8 DADA to send credit card

Potentially, DADAs could be used not only in the TownMap, but also in schematic forms within traditional user interfaces. A graphical representation of the user, the service provider, and third parties could then allow for direct manipulation of its individual graphical constituents. While both these forms

of drag-and-drop disclosures have not been implemented within the PRIME project, one might hypothesise that they can help in alleviating one problem encountered in different usability tests, namely that a few users did not really distinguish between their computer (user side) and the Internet at large (services sides). Microsoft's Internet Explorer seems to be 'the Internet' to them, and it is not obvious to them that there is a local data repository under their control with personal data and attributes.

20.5.2 Enhancing Transparency

With the diffusion and implicitness of surveillance technologies and digital data processing and storage in the modern societies, the right of informational self-determination becomes more and more endangered. Hence, the privacy principle of transparency of personal data processing is not only of key importance for the data subjects but also for a democratic society as a whole. For this reason, the EU Directive 95/46/EC provides data subjects with information and access rights. In addition to information rights guaranteed by Art. 10 of the Directive, Art. 12 grants every individual the right to access, i.e. the right to obtain from the data controller without constraint at reasonable intervals and without excessive delay or expense a confirmation whether data relating to him are being processed and information at least as to the purposes of the processing, the data concerned, and possible recipients or categories of recipients. Moreover, pursuant to Art. 12, every individual has the right to ask for rectification, erasure, or blocking of data concerning him/her as far as the processing does not comply with the requirements of the Directive, in particular when the data are incomplete or inaccurate. Furthermore, Art. 14 ensures that individuals can object, on request and free of charge, to the processing of their personal data, e.g., for direct marketing.

Users must know what rights they have and understand them in order to exercise their rights. In the PISA project, these privacy principles were translated to the HCI requirements that users are conscious of their rights, and that they understand and can exercise their rights.

20.5.2.1 Data Tracking

Being able to track what data was disclosed, when, and to whom, is an important feature for increasing the transparency of personal data processing, and is also a prerequisite for users to subsequently exercise their rights. Within PRIME, this history function is implemented in the Data Track (see also [PFHB06]). It provides the user access to transaction records, but also enables him/her to detect that the current use of personal data by a particular service provider is not in accordance with their joint agreement on a privacy policy or legal requirements, agreed upon at the time of data disclosure. PRIME usability tests have shown that people are normally not aware of their rights to rectify, erase, block and inspect data about themselves that
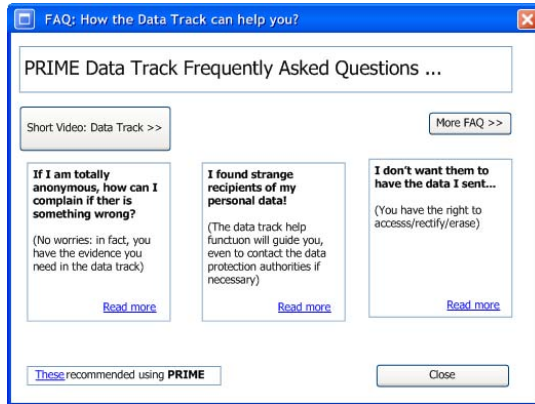
Fig. 20.9 FAQ buttons for quick access to assistance functions

companies and authorities have collected. Also the Flash Eurobarometer survey of 2008 showed recently once more that only a minority of European citizens know that they enjoy all those rights pursuant to Art. 10, 12 and 14 of the EU Directive 95/46/EC. Besides, the national Data Protection Authorities (DPA) were relatively unknown to most EU citizens [Eur03, Eur08]. The Data Track can be expanded by incorporating features that help raise user awareness in this respect and help them actively effectuate these rights and also provide them with contact addresses for help, for instance the URL of the DPA (see Figure 20.9).

As people engage in many transactions, which may involve multiple providers simultaneously, the implementation of a usable Data Track is difficult from an HCI perspective. Providing users with easy-to-use tools for finding relevant records about past data disclosure is one example. In PRIME several ways have been considered: (1) Sorting step-wise by categories, such as 'personal data' and 'receivers'; (2) Simple search box. These first two approaches are somewhat unsatisfactory because the general user is unaware of what the system does as revealed in user tests. More suitable methods include: (3) Template sentences which put search boxes within meaningful frames: "Who has received my [drop-down list with data]?" (4) A scrollable transaction track that shows all the records at once. The records are shown in abbreviated form as small pages stacked along a timeline (see Figure 20.10). A slider provides the possibility to highlight an individual page in the stack. In this way, users could browse through the records without having to understand sorting or to articulate refined search requests. Obviously, this method seems more appropriate for the beginner whose amount of transaction records will be limited. For the more advanced user, combinations of methods have to be explored and developed.
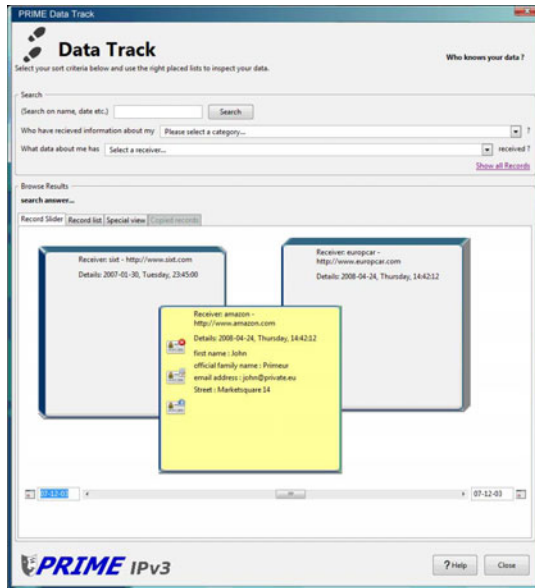
Fig. 20.10 Data Track window including template sentences and scrollable tracks

20.5.2.2   Support of Individuals in Exercising Privacy Rights

As mentioned above, individuals are usually not aware of all their privacy rights. And even if they are, they rarely exercise them because it means much bureaucratic effort to find out whom to address, to compile a letter, often to be personally signed on paper, to send it, wait for an answer, write reminders etc. When using pseudonyms (e.g., from an identity management system), this may even be more complicated because the data controller needs a proof that he communicates with the specific pseudonym holder.

Information about the individual's rights has to appear in the privacy notices (i.e., if multi-layered notices are used, it should appear in the condensed privacy notice or in the short notice if this is necessary for guaranteeing fair data processing).

Furthermore, the interface should provide obvious tools for exercising the individual's rights. It should be possible for the individuals to exercise these rights both on-line and at the physical address of the controller (see also Chapter 2 of [FHP04]), which has to be provided in the privacy notices and can be used by the individuals as a fallback solution in case that the online functions do not work.

As mentioned in section 20.5.2.1, the Data Track function also informs the users about their rights and provides access to online functions helping users to exercise these rights. Once the user has "tracked" specific transaction
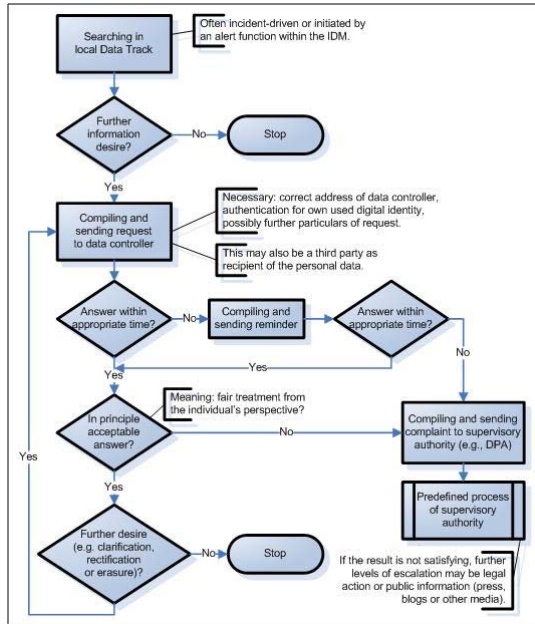
Fig. 20.11 Steps to be taken and support that is provided for exercising user rights

records, the Data Track user interface provides buttons that the user can click for activating such online functions (see Figure 20.10).

Figure 20.11 depicts the steps to be taken and support that is provided by the PRIME user-side system for exercising user rights. When exercising privacy rights, the requests have to be sent to the data controller. If there is no answer or no satisfying answer, the next level of escalation is the supervisory authority which has to be established according to Art. 28 of the Directive. This is typically a national or regional DPA.

Within a fully PRIME-enabled scenario, the right to access, rectify etc. your data even under (authenticated) pseudonyms could be realized online. But without the automatic service support, the identity management system could at least help in finding out about the address of the data controller (from the privacy policy), generating request letters, giving the needed authentication (even if a pseudonym is used), monitoring the complaint status, compiling reminders, and – in case of problems – addressing the supervisory authority in charge (see Figure 20.11).

20.6   Challenge IV: Mediation of Trust

"Trust is important because if a person is to use a system to its full potential, be it an e-commerce site or a computer program, it is essential for him/her to trust the system" [JEL03]. Usability tests of PRIME early user-side prototypes and mockups and other user studies have shown that users are often lacking trust in privacy-enhancing identity management, even though the technology might be perceived as usable. Our research has investigated the challenges of communicating to a user trustworthiness of a client and services-side systems and assurance of services-side services used to process personal data, focusing on covering assurance control, obligation management and help functions for "worried" users. For approaching this problem, an interdisciplinary approach has been taken to investigate not only the technical options but also the social factors and HCI aspects for influencing trust. The model of social factors of trust which was developed by social science researchers in PRIME and presented in [ACC+05] suggests that trust in a service provider can be increased if procedures are transparent, reversible, and – in case of breaches of trust – there are means of redress. Transparency for end users is provided by the Data Track. Moreover, the Data Track also incorporates features that help raise user awareness of their rights to access data and to request the rectification, deletion and/or blocking of their data and help them actively effectuate these rights (see above) and also provide them with updated information on consumer organizations and/or DPA that can help with legal issues. The social studies on trust factors have also shown that trust in a service provider can be increased if the user feels in control of the application. Besides, on the so-called institutional layer, trust can indeed be influenced by compliance check functions that allow users to make judgments about the trustworthiness of the services side's IT system based on evidence such as privacy seals issued by trusted independent parties or reputation metrics. We have further developed and evaluated UI proposals for assurance control for verifying whether the receiving services side still has a "good reputation" as well as a "good" privacy seal and for obligation management for increasing end-user control[2].

The scope of the "Assurance Control" which was developed in PRIME goes beyond what end users may digest – it could in principle provide service providers with advanced tools for checking out subcontractors, and also Certification Authorities can use it to check certified services. We have, however, slimmed down the assurance control for ordinary end users to rely partly on other parties performing the more advanced checks. The Assurance Control (or "Privacy Functionality Check" as we called it in our mockups and tests) user interface has been structured into three layers displaying a short status view, a compressed view displaying the overall results within the categories "Has a good privacy seal", "No blacklisting", "Provides tamper-resistant protection of data", "Supports PRIME functions", and a complete view showing

---

[2] Assuming however that services sides are "good-willing" and correctly enforcing obligations dictated by the end users that they have agreed upon

Fig. 20.12 An example of an assurance evaluation dialogue

the results of sub categories (see Figure 20.12). In a series of pilot tests, our test participants in principle understood and liked the idea of the functionality check even if well-known brand names and non-foreign vendors were clearly preferred in the tests. The whole idea of being able to get third-party judgments on an unknown site was appreciated as a means to judge the trustworthiness of a services side.

In extension to assurance control, if enterprises can be regarded as trustworthy by an end-user, individuals might also appreciate the option to be able to dictate constraints, expectations and duties to "good willing" enterprises via an obligation management system as developed by HP Labs within PRIME (see, e.g., [Mon04, Mon05]), and by this having more control over their released data. Usability test were conducted for obligation management system mockups where test participants had the option to set conditions for data use when they provided personal data via the web. The experiment showed that Internet users were able to use an obligation management system, and can be interested in such a function even though it actually increases complexity; it seemed to give the test participants a sense of being in control [PFHPCM06].

20.7   Outlook

In this chapter, we have illustrated that implementing usable and privacy-enhancing identity management systems is a challenging task and we have shown how the HCI work in PRIME has addressed some of the major challenges. Still, there are many problems to be solved and open issues that remain. In this section, we discuss some of them and provide an outlook on interesting future work.

20.7.1   Disclosing Data Using Anonymous Credentials

In mockup-based user tests we have found out that people easily mistake anonymous credentials based on some ordinary credentials such as digital passports to contain as much data as the source credential. Short pre-test information on anonymous certificates does not seem to influence this perception. For instance, in a post-test interview, one participant mentioned explicitly that his/her personal number found in his/her passport will be sent to the service provider, even though the window asking for data release stated the request as Proof of "$age > 18$" (built on "Swedish Passport"). However, the same person appreciated using electronic cash (a form of anonymous money issued by some bank or credit institute) as this procedure did not involve the credit card number or any other personal data. Possibly, the data release window could be more explicit as to the process of deriving anonymous proofs from electronic identity cards (for instance: "build proof on parts of the certificate "Swedish Passport"). Different icons and combinations of icons (such as a 'proof' icon combined with the PRIME mask to symbolise 'anonymised proof') can perhaps strengthen this. Another approach (and somewhat more screen space consuming) is to have the complete set of passport data shown but shading all the data that will not be disclosed.

However, there is also another problem connected to the use of anonymous credentials: even if better-designed information will make it possible to make people understand how little information is sent in an interaction using an anonymous credential, it has to be understood that for passports, the citizenship of the holder is always derivable if it is possible to infer which government issued the passport from which the anonymous credential was derived. Thus, the metadata that makes it possible to check the assertion of an anonymous credential may also reveal information about the person using them. This must also be made clear to users (and logged in the Data Track).

Also, more research on mental models and appropriate metaphors for anonymous credentials is needed.

20.7.2   Notification about Incidents

In addition to functions for transparency and assurance evaluations, we also see the utility of mechanisms that inform users about security and privacy

incidents, especially if they might influence (re-)use of partial identities. Specific information can be offered by feed providers, e.g. via RSS, dealing with security and privacy information on incidents concerning protocols, applications, cryptographic algorithms, communication partners, or, indeed, any PRIME-enabled software. Users of the PRIME system could subscribe to one or multiple RSS feeds which are regularly polled by the user's PRIME system. Information from the feeds which is relevant for the user is stored at the user's side and displayed: (1) when the user is going to disclose data ("PRIME Send Data?" dialogue), (2) in the "Data Track" dialogue to understand potential risks related to former transactions, (3) immediately in alerting popups when the PRIME-enabled software being used is vulnerable itself. In addition, if the information items contain dates when the vulnerability started and when it was discovered, this is helpful when interpreting former transactions which happened before the incident was known. Furthermore, the warnings should not only comprise mere information on the incident, but also ways how to overcome or at least deal with the vulnerability. To ensure authenticity of the provider's feed items, they are digitally signed, and the signatures are checked in the polling process. The provider's public key has to be integrated at the user's side feed management component.

20.7.3 Linkability Computation

The information available in the Data Track in principle allows the user to find answers to questions such as "If company A and B pool their customer databases, what can they infer about me?" The Data Track should thus be extended with the capability to do compilations like that and also simulations of linkability based on released personal data. For novice users, it could take the form of simulation games. Other linkability computations can be based on using available resources for computing the likelihood that someone else has one's name within this district of this city, etc. Naturally, such computation would be most beneficial during a data release action, but having the possibility to do it in the Data Track together with other linkability computations may allow the user to better understand linkability risks. Clauß [Cla07a] has made a thorough analysis of linkability including usability options, which however still need to be implemented and tested.

20.7.4 How Ontologies Can Be Utilised for UI Design

Ontologies allow gathering data and data types under common denominations, such that 'name' consists of 'prefix', 'given name', 'middle name', 'family name', 'suffix'. Other name categories can be constructed as well as, for instance, different address structures. For HCI it is of particular relevance that the ontologies can be extended to host tooltip explanations and translations of data types into different languages. Automatic translations are very

promising also in a broader scope where privacy policies and (end-user's) obligation settings are considered. Standard messages would make it possible to automatically translate between the languages of the service provider and the customer, providing for informed consent in the best possible way. Translatable standard messages would also be of great advantage for the use when contacting the data controller via the Data Track's assistance function as discussed in 20.5.2.1.

Naturally, it is hard to plan all possible ontology structures before releasing a fully-fledged PET system. For future extensions of the PRIME architecture one may address the capability of the system to dynamically include external ontologies. This presupposes a trusted framework to guarantee the consistency and correctness of the ontologies. This also requires some public key infrastructure (PKI) to certify and prove the integrity and authenticity of the external ontologies.

21

Technology Assurance

Tobias Scherner and Lothar Fritsch

JWG University Frankfurt

21.1 Introduction

This chapter documents the experiences of assurance evaluation during the early stage of a large software development project. The PRIME project researches, contracts and integrates privacy-respecting software to business environments. There exist several approaches to ensure the quality of secure software. Some of these approaches have the focus of quality assurance at a very early stage of the development process and have weaknesses to ensure the quality of this process until the product is ready to enter the market. Other approaches, like the CC, focus on inspection, or more concrete evaluation, of ready-to-market products.

While assurance evaluation with ISO 15408 Common Criteria (CC) within the certification schemes is done after a system has been completed, our approach executes evaluation during the early phases of the software life cycle. The promise is to increase quality and to reduce testing and fault removal costs for later phases of the development process. The first results from the project suggests that the Common Criteria can define a framework for assurance evaluation in ongoing development projects.

Our approach aims to bridge the gap between requirements engineering, code production and post-evaluation. This is motivated by two effects we expect: First, faults discovered earlier can be removed faster, and second, these discovered weaknesses can be removed cheaper. For making this point clear, we first have a look at testing, verification and validation literature from the software engineering field on knowledge. Then we will briefly introduce the Common Criteria scheme. Following this, we describe our process approach to detect security assurance problems in the ongoing development process. We

tried to introduce an inspection process that is inspired by the CC evalua-
tion scheme to earlier phases of the software engineering process. In the end,
we provide insight on how the evaluation process has been applied into the
PRIME-project.

21.1.1   Cost of Testing

First, we will deal with the question whether early testing efforts in secure
software development are economically justified or not. Early testing intro-
duces cost into the design phase - and it might not be trivial to find evidence
whether it is worth the investment.

In the literature, one can clearly identify that early fault removal is more
economic than late fault removal. Although on first sight, one might conclude
that early testing and validation simply shifts testing cost to designers and
developers, some economic evidence exists that due to network externalities,
code re-use and the software engineering process, early failure detection is
notably cheaper than later failure removal. In [Esk01], the cost of fault removal
during different phases of software engineering increase exponentially as listed
in table 21.1.

Table 21.1 Cost of fault removal in software engineering according to [Esk01]

Phase	Cost
Requirements	10 $
Analysis	20 $
Design	30 $
Code	50 $
Testing	200 $
Install	800 $
End User	1500 $

Here, early fault removal clearly is much cheaper than later fault removal.

An economic model of bug removal is constructed in [Vie95], where the
authors gather evidence for the argument that early bug removal is more
efficient than later testing and removal.

We looked at several approaches to deal with testing. The United States of
America National Aeronautics and Space Administration (NASA) hat a strict
standard on software quality [Gre92]. In section 3.2.1.2.1 of the document,
the mission of software assurance is defined in this way: 'A strategy that
emphasizes prevention, not correction'.

In [Exl04], a consulting firm suggests to use CC elements for early software
validation due to the fact that the CC provide a large variety of standardized
information and processes on security vulnerabilities. An example of using the
CC during a software development process can be found in [VWW02], where

a Palm pda software has been developed using a process based on the CC requirements.

21.1.2   Common Criteria

The Common Criteria for IT Security Evaluation, short CC, provide a collection of generic components of security requirements to aid in the specification of product or system security attributes. The version 2.3 is similar to the ISO (International Organization for Standardization) standard 15408. The traditional utilization of the CC is the usage as the basis for evaluations of security properties of IT-systems and software. The main objective of the CC, besides a well known and excepted standard, is the evaluation of products. This can, among other purposes, be used to provide users and customers a decision support base if this evaluated object meets the own requirements, or requirements that have been investigated by experts for supporting the development process and providing guidance to customers whether the product meets their needs. Examples for evaluated Products are Smartcards from the credit card sector.

The CC advise to produce Protection Profiles (PP) and Security Targets (ST). PP's are an implementation-independent set of security requirements for a category (application specific) of Target of Evaluations (TOE) that meet specific consumer needs. On the other hand ST's are an implementation-dependent set of security requirements and specifications used as the basis for evaluation of the identified TOE. An ST can be compared to the corresponding PPs to assess whether the postulations of the PP are met.

Preferably, the CC shall support the developers to meet the postulated requirements right from the beginning of the development process. But until now this policy is not a formal defined part of the ISO 15408 standard.

21.2   Early Security Validation with CC

Our approach is to adapt the principles of the CC of building PP's and ST's during the development process without the standardized components of the CC, but properly reflecting the security requirements which have been defined for the project results. The comparison of ST and PP already during the development revealed different lacks which have been reported to the developers to solve the problems until the next evaluation loop. From the perspective of the project, the early involvement of evaluators offered the chance to fix problems with a lower cost, effort and to fulfill the high self-expectations and the expectations of the commission as well as future users.

21.2.1   Evaluation and the Common Criteria

The basis of the evaluation process is the, at the live time of the project, official version 2.3 of the Common Criteria (CC, IS 15408). Essential for developers is

the reading of the 'Common Methodology for Information Technology Security Evaluation' [fSI07b]. This document describes the methodology of different evaluation assurance levels (EAL) including lists of necessary activities.

Following the methodology of the C,C the assurance through evaluation has several meanings, and the following list can be seen as a basis of the CC evaluation [fSI07a]:

- analysis and checking of process(es) and procedure(s);
- checking that process(es) and procedure(s) are being applied;
- analysis of the correspondence between Target of Evaluation (TOE) design representations;
- analysis of the TOE design representation against the requirements;
- verification of proofs;
- analysis of guidance documents;
- analysis of functional tests developed and the results provided (by the software developer);
- independent functional testing;
- analysis for vulnerabilities (including flaw hypothesis);
- penetration testing.

The process of the evaluation is an integrated process over the whole life cycle including the planning of a software project, developing and integrating of components, installing and using the software. So, the above listed elements of an evaluation are far from being complete, but the different evaluation assurance levels extend the evaluation basis by the assurance aspects described in [fSI07a].

The evaluation of the project components is not bound to certain evaluation levels and all the formal regulations, but developers and evaluators have to agree on a defined level. From the evaluation point of view the general conditions should follow the requirements of the evaluation level 4. This recommendation is caused by the project technical design principles that state very clearly that the maximum of privacy shall be achieved and to ensure that the principles are fulfilled we need a high level of assurance.

However, the discussion about which level of assurance is needed has been intensively discussed between evaluators and developers and due to the research character of the project, the applied evaluation level has been lowered for being able to have a match of what pre-mature technology could provide and what a reasonable evaluation could provide to the development and the project itself.

21.2.2 Basic Preconditions for an Evaluation

This section describes the basic requirements for an evaluation of software in general, but focussing on one of the main results, the integrated prototype with its development cycles. Under the notion 'precondition' we summarize all

documentation that an evaluator needs to accomplish a basic evaluation process in an integrated manner like it is described above. The following sections describe in detail which documentation an evaluator does normaly expect for:

- Implemented security functions.
- Threat analysis, security objectives, strength of the implementation.
- Test plans.
- Best practice examples for the application prototype on how to use the provided interfaces.

21.2.3 Implemented Security Functions

An evaluation normally requires a list of the implemented security functionalities. This includes on the component level a list of what kind of security functionalities are implemented including the specification (e.g. kind of encryption algorithm, description of the distribution of the keys and the storage), which countermeasure is aimed to protect against what kind of threat in which expected strength. On the level of the prototype, a description of the interaction of the different components is mandatory.

21.2.4 Threat Analysis

Threat and vulnerability analyses are one of the most important parts of the preparation material for an evaluation. The aim of vulnerability analysis is to find weaknesses of the security of a system or parts of the system. The threat analysis is based on the perceptions of the vulnerability and characterizes the possible effects of the found weaknesses. The documentation empowers the evaluators to understand the background of implementations and to come to an assessment whether the known possible threats can be counter measured by the implemented security functions. Following the CC part 3 [fSI07a] vulnerabilities can arise through failures in:

- Requirements – that is, an IT product or system may possess all the functions and features required of it and still contain vulnerabilities that render it unsuitable or ineffective with respect to security;
- Construction – that is, an IT product or system does not meet its specifications and/or vulnerabilities have been introduced as a result of poor constructional standards or incorrect design choices;
- Operation – that is, an IT product or system has been constructed correctly to a correct specification but vulnerabilities have been introduced as a result of inadequate control upon the operation of it.

A possible, and from our point of view, adequate presentation of a threat analysis can be found below in tables 21.2 through 21.4.

Example: communication

Table 21.2 List of components

Component's name:	Component's number:	Interacts with the following components:	Description:
Communica-tion	C_1	Event man-ager	Responsible for the communication between the users, service providers and internal communication.

Table 21.3 List of threats

Number of the threat:	Description:
T_1	Communication can be eavesdropped (and analysis provides meaningful results).
T_2	Communication partners can be revealed to a third party
T_3	Communication can be altered
T_4	Communication partners can forge their identity.
T_5	. . .

Table 21.4 List of security objectives

Number of security objectives:	Description of security objectives:	Eases impact of threat number:	Strength: (low / medium / high)
CM_1	Use of encryption mechanism like 3DES and AES	T_1, T_3	High
CM_2	Use of authentication mechanism like certificates	T_4	Medium
CM_3	Use of Mixes and dummy traffic	T_2	Low

21.2.5   Test Plans

Test plans have multiple dimensions. The first dimension concerns the components, the integration and the system as it is for example described in [RM04]. Each of these levels has to be tested and the tests have t be documented. The second dimension covers the testing of security functionalities, tests of the interfaces to later on used parts of the project and handling of unexpected situations (e.g. test of stability of the programs if these programs are contacted with unexpected enquiries).

The documentation of the tests covers:

- The type of the conducted test (e.g. functionality, security or stability test).
- Scope of the test (e.g. tested components, interaction with other parts of the project software).
- The documentation of the test procedure. This includes the test configuration including the used tools and the underlying infrastructure inclusive test criteria and conditions that describe why tests have been terminated.

21.2.6   The Documentation of the Test Results

A suitable test standard is the IEEE standard '829-1998 IEEE Standard for Software Test Documentation' [IEE98] which accurately describes the composition of test plans and offers standardized documents to support the efficiency of the test team and additionally the evaluators.

21.2.7   Evaluation Process

We evaluated the various versions of integrated and applications prototypes by using the following, newly developed evaluation schema.

21.2.7.1   Process One

The starting point of our evaluation is the test release of the to be evaluated software deliverable. It provides an overview of the included security and privacy functionalities. For each component, the evaluators have to examine its contribution to privacy and security protection.

This contains in detail:

What is the purpose of the component (e. g. what the benefit of the implementation for the end-user is)? The main sources for this are the project's technical deliverables.

What are the possible threats? This has to be analyses by an independent threat analysis based on input from the developers. For the most privacy-protection goals, there normally exist several threats. Hence, we want to summarize how the targeted benefit of each component can be weakened or totally neutralized through different threats. This detailed analysis considers the fact that a system is only as strong as its weakest part.

For the last two items, one has to rely on input provided by the developers of the components, who have to provide their threat analysis and security objectives as described above. The approach of creating an own threat analysis leds to a better understanding of the to be evaluated software. The next step is to analyze the specifications. The purpose is to evaluate if the provided functionalities can deal with the investigated threats. This results in a first indication of whether the prototype fulfils the claimed requirements or not. To be able to compare the investigated requirements one has to build a security target (ST) for the integrated prototype.

21.2.7.2   Process Two

Starting from the requirements postulated in the various requirements deliverables, the evaluators have to summarize and structure the requirements regarding the to be evaluated prototype.

Further on, the next task is to create a lightweight Protection Profile (PP). The notation 'lightweight' was chosen, because the approach does not necessarily fit into the formalized requirements of the Common Criteria given that the postulated requirements would have to be transformed one-to-one into the structure of functional components of the CC. However, the lightweight PP reflects the basic requirements [Pro04] like unlinkablity, pseudonymity, and anonymous communication in natural language and it provides a TSF (TOE Security Functionality) description according to the CC.

21.2.7.3   Joint Process

To combine the two previous parallel processes the evaluators have to compare the Protection Profile of the users' point of view and the security target of the components. At this point the evaluators have to analyze in how far the postulations of the Protection Profile meet the requirements of the security target. This operation can be understood as a mapping of the two constructs. Due to the deviation of what the lightweight PP stipultes and what is included in the formalized requirements of the CC, the mapping is more a global examination whether the ST claims conformance with the PP than a real conformance check. At the end of this joint process, it is possible to get to conclude about the quality of implementation of the integrated prototype.

21.2.8   Experience with CC-Based Project Evaluation

The experiences with the evaluation approach where directly linked to the progress of the project and to the degree of dependency of the prototype and the amount and complexity of the used PRIME-components.

21.2.9   Integrated Prototype

The first cycle of the assurance evaluation of IPV.1 could not be performed due to several reasons. First, the analysis showed that the discrepancy between the required and available documentation was too high. An investigation of this phenomenon revealed that developers and evaluators had a different view on what an evaluation is. This is a commonly observable problem while dealing with teams consisting of specialists coming from different domains and cooperating in large projects. One approach is to use a prototype as a boundary object for coming to a common understanding of the requirements regarding the prototype [GHNR04]. Building a boundary object for evaluations could

be a great chance for the project to reach to consent about the scope and to agree about the boundary conditions of evaluations within the project.

Moreover, the assurance evaluators detected discrepancies between different statements provided by the developers of component and the integrators of the components about the stage of implementation. This problem seems to be caused by two associated circumstances. The original root were integration problems which resulted in deviations from the integration time plan. Thus, the deviation created stress and inhibited adequate communication between component developers and integrators. Thereby, the component developers had no updated information whether their component was integrated or not.

Secondly, the implemented security functionalities of prototype version 1 were not as fully implemented as it would have been necessary for a successful assurance evaluation.

Of primary importance were the questions how to deal with the inaccurate documentation and the lack of important security functionalities. Facing these problems, the assurance evaluators came to the decision of suspending the evaluation process and instead starting to prepare the evaluation process of version 2, and educating the developers better about assurance preconditions. This approach was fruitful and thus the evaluation of IP V.2 was successful in many terms. The documentation was well prepared even it was spread over many sources and available in different versions, depending when the documentation has been prepared. The claimed privacy-preserving functions where implemented and fulfilled the specification. The maturity of second version of the prototype was significantly increased and was convincing in terms of the assurance evaluation. Suggestion for security modeling, security concept documentation and threat analysis have been made in the PRIME document "Guidelines for assurance evaluation" (Version 0.9, 21-Oct-2005).

21.2.10   LBS Prototype

The experiences gained with the LBS prototype were different compared to the integrated and the eLearning prototype. Specializing on LBS, narrowed the set of required PRIME-componets and lead to clear and good results in both of the assurance evaluation cycles of the LBS-prototype. Documentation as well as proof of functionalities of the prototype were best-practice examples of how to build privacy respecting prototypes. Finally, one could conclude that PRIME has successfully transferred knowledge from fundamental research to the product level in a very specific area, which are at the same time suitable to many similar use cases in the world of LBS.

21.2.11   eLearning Prototype

The eLearning and later on, the collaborative eLearning prototype used some more PRIME-components compared to the LBS-prototype which resulted in
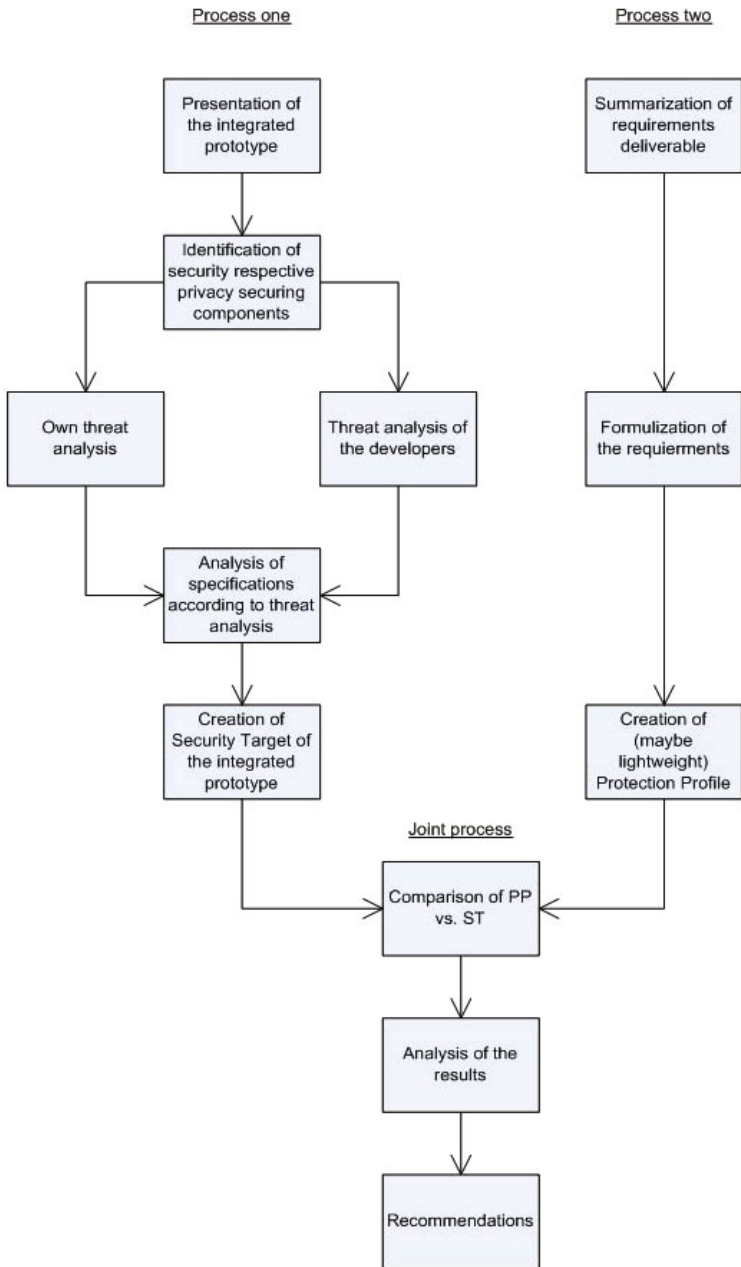
Fig. 21.1 Proposed evaluation process

an increased complexity compared The late release of the underlying integrated prototype lead to race conditions in meeting the projects timeline, which again lead to shortcomings of the implementation of the eLearning prototype. Due to the wide scope of the chosen scenario, users had the freedom to put as much privacy-sensitive data into the system as the wanted. Since the PRIME technology was not intended to control such kind of intestinally left data; the assurance of privacy-preserving functionality of the eLearning prototype was primarily limited by the users' action. However, the handling of data was ex-post not transparent to users and they had blindly to trust the system that it does what the specification promised.

21.3   Conclusion

The main conclusions of the iterative assurance evaluation process is that developers had difficulties to meet the expectations of the evaluators, especially in the beginnig of the project. To some degree, this was caused by dynamic development processes in a research environment and partially by other factors that are inherently included in large software development processes. At the beginning, some components had nothing but a claim about their security functionality, and no documentation useful for an assurance evaluation. Developers missed to document their threat and risk analysis and had to face many integration difficulties which resulted in shortcomings of the integration into the overall concept. The lack of communication among the developers on the one hand and between developers of the components and system integrators integrators on the other hand had a strong impact on the evaluation result. The suggestion was that the developers have to follow a more formal process regarding analysis, specification, developing and documentation. In the following development cycles the developers had a higher degree of reflection on their work to discover inconsistencies during their decisions. One major conclusion is that without applying our evaluation approach, we would not have found many problems at the early stage of the project. A traditional CC evaluation would have brought up these problems at the end of the project, which would have endangered the success of the whole project beyond its deadline.

Our first application of the CC-based early evaluation process discovered many design and documentation inconsistencies and surfaced several implementation problems. It therefore can be regarded as a success. After our next step—education of developers about accurate analysis and documentation— the results in the next evaluation cycle were satisfying in many means. It provided insights in the usefulness of our evaluation process and lead to improvements. The results suggest that our evaluation process supports early security fault detection and removal, which according to section 21.1.1 will lead to lower cost of the software engineering process.

22

Requirements for Identity Management from the Perspective of Multilateral Interactions

Stefanie Pötzsch[1], Katrin Borcea-Pfitzmann[1], Marit Hansen[2],
Katja Liesebach[1], Andreas Pfitzmann[1], and Sandra Steinbrecher[1]

[1] Technische Universität Dresden
[2] Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein

22.1 Introduction

22.1.1 Objective of the Chapter

In this chapter, application scenarios for identity management systems will be discussed in order to identify requirements which should be or already are considered in the design and implementation of privacy-enhancing technologies (PETs).

There will be a comparison showing that the current view on identity management within bilateral scenarios – as mainly addressed by PRIME – should be broadened. Further application scenarios need to be covered in which users

- interact in arbitrary numbers and by different technical means,
- establish various kinds of potentially reciprocal relationships,
- generate and exchange personally identifiable information (PII) which needs to be protected.

We will call such scenarios *multilateral interactions*. Example scenarios in this area are motivated by the collaborative eLearning prototype that has been developed in the context of the project PRIME and focuses on multilateral interactions within a collaborative eLearning environment.

Results achieved by PRIME will be documented, and open issues for further work in the area of identity management for multilateral interactions will be indicated.

22.1.2 User-Controlled Identity Management: From Chaum to PRIME

Chaum [Cha85b] laid a basis for user-controlled identity management with his influential work on digital pseudonyms, blind signatures, and anonymous credentials. In accordance with traditional client/server applications, Chaum splits the world into organisations and individuals. Individuals – represented by their client devices – disclose data to several organisations – technically represented by servers. The individuals want to protect themselves against surveillance through information sharing between those organisations. Organisations on the other hand want to secure their resources from unauthorised access and misuse by individuals. The use of various self-generated unlinkable digital pseudonyms by an individual in interactions with various organisations prevents those organisations from exchanging information about that individual. Digital signatures assure integrity, authenticity, and accountability. Credentials offered by distinguished organisations are digitally signed proofs of attributes or rights of an individual that can be shown in any interaction with other organisations. Anonymous credentials help on the one hand to ensure confidentiality of content, since it is verifiable that only authorised individuals and organisations get access; on the other hand anonymous credentials offer anonymity for the holder of the credential. Chaum describes scenarios where single individuals have relationships with several organisations and exchange data with them. Organisations communicate with each other in general, however since no common identifiers exist for individuals, organisations are not able to share data about individuals.

In 1992, Chaum extended his model of the world with tamper-resistant modules (a sort of electronic notary) on the users' client system [Cha92]. Servers of organisations trust these modules at least to some extent regarding the integrity of data of an individual, so enabling different kinds of off-line services; however, the tamper-resistant modules do not have any effect on privacy. In the context of the PRIME project, Chaum's ideas were resumed and developed further. The focus lies on client/server-based scenarios between users and service providers. In these scenarios, users' PII need protection against service providers. The "users" in the PRIME model correspond to the "individuals", and similarly the term "service provider" replaces "organisations". In PRIME, requirements for identity management systems are identified and evaluated for selected scenarios. Based on these requirements, solutions are developed and implemented in different prototypes. Unlike Chaum's approach, PRIME includes an option for tamper-resistant modules on the service providers' systems, too. Such modules offer technical support to assure the trustworthiness of service providers by checking policies or trust seals which can be issued

by external parties such as data protection authorities. This assurance serves as a basis for a user's decision whether and which PII she discloses to the service provider and for what purpose [Ber07]. Additionally, the server could inversely check the client's trustworthiness.

22.2 Multilateral Interactions Using the Example of a Collaborative eLearning System

22.2.1 Multilateral Interactions

Social and collaborative software requires the consideration of several users interacting with each other as well as with one or more service providers. If potential reciprocal effects of these relationships – regardless of their kind – are taken into account, we speak of multilateral interactions (MLI). Communities on the Internet are a common example for multilateral interactions. Another such example is the *PRIME collaborative eLearning (CeL) prototype* that offers individual learning supported by a technical system (user - service provider interactions) as well as collaborative and cooperative learning via a technical platform (multiple users interacting). It therefore fosters building of learning communities – so-called "Communities of Practice" [Wen96]. Through the incorporation of multilateral interactions, modified and extended requirements for privacy-enhancing identity management systems arise. In the following sections, these requirements are identified and further specified.

22.2.2 Stakeholders

Internet communities are examples of an MLI environment. Stakeholders of such an Internet community (based on a client/server architecture) can be identified and grouped into

- provider(s) of the technical platform,
- community content operator(s), who provide a framework for user-generated content,
- users, who produce content themselves.

Figure 22.1 visualises these stakeholders and their relationships. Community content operators and users interact on application level as well as build up and maintain multiple relationships. Furthermore, relationships between providers of the technical platform and all other users and content operators exist; however, these relationships are not considered in the following. For simplification, it is assumed that technical providers do not play a role for the view on privacy and identity management on *application level*. With some effort, this could be realised by using standard security mechanisms.

Since in communities – as an example of an MLI area – contents as "provided services" are created and further developed by users themselves [Rhe93],
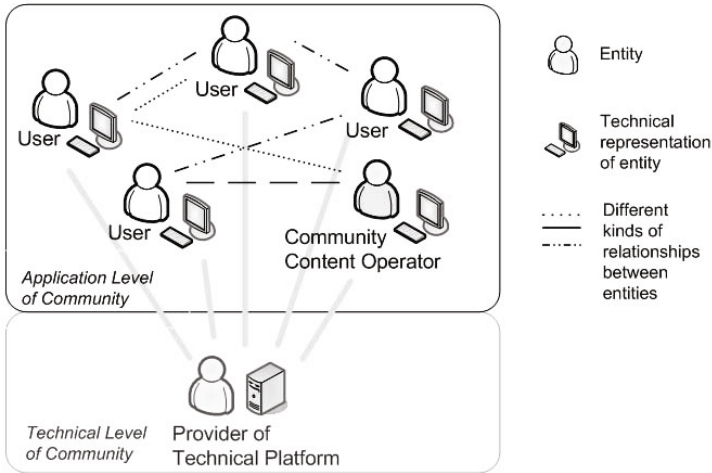
Fig. 22.1 Players and relationships in an Internet community

it is not possible to differentiate between service provider and user as can be done in traditional scenarios. In the context of MLI research that focuses on multilateral scenarios on application level, therefore we speak of *entities*. The term "entity" comprises individuals as well as organisations that both could provide and offer services. On the one hand, it is assumed that each entity has possibilities to capture, process, and store data of others; on the other hand, all entities want to control and protect data related to them. Interests of a single entity regarding its privacy protection differ according to special situations and kinds of relationships with other interaction partners. However, these interests need to be well-coordinated and well-managed by the individual.

Within a collaborative privacy-enhancing eLearning environment such as the PRIME CeL prototype *BluES'n* (*BluES like universal eEducation System privacy-enhanced*, cf. Chapter 24), from an administrative point of view, entities act as guests, participants, or owners of so-called workspaces. Workspaces are used as metaphors for structuring the eLearning environment on a semantical basis into several sub-areas, which are characterised by functionalities, resources, and users assigned. Besides *administrative roles*, the eLearning system provides *functional roles* which offer specific features to an entity according to a certain situation. They equip the entity with selected rights and imply expectations on the behaviour of the entity in this specific role [BPL07]. Through the exchange of information between entities and collaborative work on and construction of shared learning contents, a learning community is formed within the CeL prototype.

Below, requirements for the CeL prototype as a privacy-enhancing MLI environment integrating identity management are described from the perspective

of the three administrative roles. It is assumed that all entities – regardless of their role – want to protect their privacy and therefore only disclose a minimal amount of necessary data. Also, there is the following distinction:

- **Guests** only have read access to public workspaces and resources. They do not take part in active collaboration and cannot be recognised by other users later.
- **Participants** want to contact users and be available for others during their work within the collaborative environment. Participants are interested in building up and maintaining different kinds of relationships with other participants and owners of workspaces and to actively take part in collaboration. They want to control whether they can be recognised in the system later, and by whom. For working within the learning environment and with other users, they need write access to the corresponding workspaces and resources.
- **Owners** administrate workspaces and their resources. The decision which PII they want to disclose may not only depend on their personal preferences, but also on rules and legal regulations. Owners want to control access to their workspaces as well as access to the resources that are created and presented within a particular workspace.

Participants of a particular workspace can act in other workspaces in the role of guests, participants, or owners. The same applies to guests and owners of a particular workspace. The set of users of a workspace could be identical to, completely different from, or overlap to some extent with the sets of users of other workspaces.

22.3 Building Blocks of a Privacy-Enhancing Identity Management System for MLI

Within the collaborative eLearning environment as an example application for MLI, entities disclose PII in several workspaces to varying extent with regard to the kind and value of these PII and the particular interaction partners. Different kinds of relationships are established between entities when working and learning together. These relationships are maintained beyond the particular workspace and may serve as base for further collaborations. It is the objective of an identity management system in traditional use cases to support the user in the generation, management, and storage of her PII. For MLI environments, it is additionally required to support relationships and collaborations based on these relationships.

Many parts of necessary building blocks for identity management can be found within the approach of PRIME. From the perspective of MLI, some extended requirements arise and should be considered for the building blocks

which are presented in general in [BPHL+06b] and [BPHL+07b]. In the following, new building blocks covering *relationship information* and *external regulations* are added, and the extended requirements for each building block are pointed out using the CeL prototype as an example.

22.3.1   Pseudonyms and Partial Identities

Within the context of a user-controlled identity management system for MLI, pseudonyms and partial identities are used to control privacy by the individual entity. Re-use of pseudonyms and partial identities provide each entity with possibilities to control and manage recognition of herself by others within the collaborative environment as well as to maintain relationships.

Depending on the context, other parties involved and objectives of an interaction, the entity should be able to decide, whether, to whom, and for which purpose it wants to disclose its personal data. Appropriate pseudonyms as identifiers for partial identities have to be chosen by the entity in order to control linkability of PII. PRIME already offers such functionalities which are needed for traditional privacy-enhancing identity management as well.

An additional requirement evolving from the perspective of MLI is the support for confidential exchange of information between entities of a subgroup that are using the application. For the implementation of this requirement through cryptographic mechanisms, the various possible subgroups and their dynamic changing of members over time by excluding or adding some entities have to be considered. PRIME does not provide such features, but they would be particularly useful in MLI environments.

22.3.2   Relationship Information

Besides PII that belongs to exactly one entity, relationship information exists that is not assignable to only one entity. Such information pertains to at least two entities and should be stored, managed, and evaluated in a privacy-respecting way as well. In contrast to PII where the intentions of only one individual have to be respected, relationship information needs to take into account preferences from all parties involved.

PRIME as an individual-centric approach for user-controlled identity management does not sufficiently consider relationship information. Mechanisms for negotiation of privacy policies between arbitrary kinds of entities are also not yet provided. Though, Data Handling Policies (DHPs) in PRIME represent a means approaching this requirement. These DHPs allow the user only to specify how her data should be handled by a server, i.e., processed, stored, and possibly passed on further on the service provider's side (cf. Chapter 11). However, from the perspective of MLI, privacy policies need not only to be specified between user and service provider, but also between any kinds of entity in order to support direct and indirect interactions between users, too.

The PRIME Data Track enables users to get a history overview of their contacts and data disclosed to servers (cf. Chapter 20). It is not possible to enter and manage relationships to any other user unless the DHP includes such information.

Examples of relationship information are address books or other lists which are created by the user and indicate whom she knows and what kinds of relationships between her and other entities exist (e.g. list of friends, list of tutors). Further attributes of relationships can be stored and managed. It should be asked how the interests of the entities, which appear in such lists, are considered and enforced. Note that information related to them personally might not only be stored locally on their devices, but also on the clients of all relationship partners. Thus, a number of research questions arise. Some of these are listed in the outlook section (22.4.3).

22.3.3 Searching for and Finding of Interaction Partners

Since collaboration and communication in MLI environments represent direct and indirect interactions with other entities, the process of searching for and finding of potential interaction partners is a privacy-relevant one. It even conflicts with the approach of user-controlled linkability. In general, the entity that is looking for collaboration and communication partners has two options:

1. It uses the public, e.g. by self-advertising.
2. It reverts to existing relationships, e.g. the friend-of-a-friend principle.

Within the CeL prototype, searching for and finding of interaction partners is supported by workspaces that serve as central meeting points for all users of the system, and by an information area where it is indicated which partial identities of other users are currently available in the eLearning environment. In order to efficiently and effectively foster collaboration and communication between entities, it is necessary to create and publish user profiles that contain selected attributes, i.e., disclose some partial identities, which can be browsed by other users. Therefore, an identity management system for MLI should provide the user features for managing such public partial identities. In this context, it needs to be investigated how to organise storage of such data. Either

- each entity decides itself how and where to store PII, or
- there are specifications imposed by the system.

Further, constraints regarding availability and trusted areas have to be considered when making decisions about the place of storage of PII that should be searchable for community members. For instance, a local storage on the user's client would leave the profile data in her trusted area, but searchers could only browse the profile if the user in question – or more precisely her client device – is connected to the network. If profiles are stored at server side, on the one hand, higher availability could be expected (assuming that

the server is always online and no technical problems occur), but on the other hand users are forced to extend their trusted area and need to trust the server regarding security and privacy of their PII. In order to increase availability in decentralised environments, redundant storage of profile data is possible on multiple clients which the owner of the PII trusts. The PRIME approach has to be extended in order to provide sufficient support for requirements that evolve from the stated issues.

Furthermore, a user has no option for intentional linkage of different partial identities belonging to that user. Such a feature would give users in MLI environments the possibility to start interactions with a great amount of privacy by using a variety of partial identities and – if trusted relationships have been established – to decrease privacy by linking at least some partial identities in order to aid collaboration with selected partners.

22.3.4 Trust Management and Reputation

Reputation and trust may decisively influence the process of searching and finding interaction partners in multilateral interactions, e.g., in the user community of the CeL prototype. Furthermore, trust and reputation influence the negotiation of privacy policies between entities. In multilateral interactions, reputation does not have to be a static value assigned to an entity. It might also be of interest who has given the rating and what kind of relationship exists between this entity and the one that gets the rating. The reputation value of an entity – or more precisely of a partial identity – represents PII and needs to be protected and managed accordingly [MO04]. One special characteristic of this attribute is the possible flexibility of its value on which the holder has no direct influence. Another characteristic is the question of transferability of reputation values between partial identities of one entity (cf. [PS08]).

The CeL prototype contains a component named "reputation management" which allows for evaluation of contents. The reputation value of a partial identity is calculated indirectly by taking into account the evaluations of contents of the eLearning environment which are created by that partial identity [Jus06]. Such a reputation value represents an attribute of a partial identity within the CeL prototype; however, it is not possible to evaluate particular users directly.

In order to provide users of the MLI environment with an individual view on evaluated objects such as content or other users for instance, the calculation of reputation values could be tailored to the needs of the user requesting the reputation value. In this case only evaluations from those other users are considered who fulfil one or more specified criteria from the requester. Such a criterion might be the relationship to the user requesting the reputation value. For the calculation of such values tailored to the needs of a certain user, it is necessary to have a listing of all evaluators whom the requesting user "trusts", i.e., the ratings of which should be considered.

In the implementation of the PRIME middleware, the aspects trust management and reputation are integrated by means of the so-called Platform Trust Manager (cf. Chapter 17). This component allows each user to evaluate the trustworthiness of service providers automatically to some extent by checking policies or evaluations from third parties such as data protection authorities. However, the realisation of the Platform Trust Manager is currently limited to traditional user/service provider scenarios.

22.3.5 Awareness Information

Entities working in collaborative environments perceive information regarding their tasks and other entities, as well as information regarding the environment itself. This kind of information is called awareness information [GGR96]. Within the interaction environment, different kinds of awareness information exist.

1. Awareness information that refers to the entity itself: its state, its activities and level of attention, etc. Thus, awareness data may represent personal information.
2. Awareness information that refers to the application as an interaction environment: available resources and functionalities, other entities, their availability and their level of interest, etc.

Since PRIME does not focus on collaborative scenarios, available awareness information is limited to certain data about the user and her actions. Users get informed about the state of the PRIME middleware, e.g., whether it is running or not. Additionally, the PRIME middleware gives the user feedback indicating which PII she discloses in interactions with service providers. An identity management system for MLI should offer possibilities to publish own awareness information in a privacy-preserving way. Further, it should allow access to awareness information of other entities in the environment while respecting their privacy policies. Thus, the creation or selection of appropriate partial identities and the establishment of relationships can be supported.

The Awareness Framework of the CeL prototype addresses these issues and enables the perception of other users in the eLearning environment [Fei06]. The implementation of the awareness component currently allows two levels: Either all awareness information regarding one particular user is available to all other users, or – by switching into a "hiding mode" – no awareness information is provided to any other user. Future developments should consider privacy further by providing more levels for disclosure of users' awareness information, e.g., by incorporating relationships.

22.3.6 Context and History

An entity in a multilateral interaction environment has various goals towards different interaction partners at different places; we speak of various *contexts*

which the entity acts in. In order to enhance privacy, personal data is partitioned by the entity according to the different contexts [FE06]. The selection of a partial identity will be made in dependency of the current context and with consideration of history data, i.e., which partial identities the entity has used in the same or similar contexts earlier and what personal data was already disclosed in this scope.

The PRIME Data Track provides an overview to the user of which data she has revealed towards which recipients and for what purpose in the past. Additionally, the Decision Suggestion Module (DSM) that was developed in PRIME supports user-controlled determination for the granularity of contexts and based on this decision it also supports automatic detection of context switches. For privacy protection and to prevent unwanted linkability of PII, the user is informed about each context switch and another partial identity can be suggested according to the configuration. The DSM is currently specifically realised for the CeL prototype, but it should be made available for any MLI environment.

The PRIME IP role model [Ber05] allows the user one role per partial identity, which is not flexible enough to fulfil requirements from the perspective of MLI. In MLI environments such as the CeL prototype, entities might act in several roles, administrative and functional ones, using the same partial identity if they want to be recognisable, respectively linkable.

22.3.7   Access Control

The use of traditional access control lists (ACLs) or role-based access control (RBAC) approaches is not possible within privacy-enhanced environments where entities dynamically switch their partial identities depending on the current situation. Thus, a more flexible access control concept is required. In the context of PRIME, an approach, which is inspired by capabilities, is suggested. In order to avoid linkability of different partial identities of an entity, convertible credentials are used (cf. Chapter 10). By means of credentials, an entity can give proof of certain attributes, abilities, etc. without being bound to a particular partial identity.

Access to resources of the CeL prototype is managed by various available access modes to the workspaces from which the owner selects one for her particular workspace, i.e., public, restricted to specific pseudonyms, by request and explicit authorisation, or by proof of property, respectively. In order to respect privacy of the users as well as to protect resources from unauthorised access, the last mentioned mode works with credentials, i.e., only holders of specified credentials get access. Motivated by the option to gain certain qualifications and certificates in the form of credentials within the eLearning environment that can also be available to users in contexts apart from the CeL prototype, questions arise concerning the export and validity of credentials outside their system of origin. Therefore, an identity management system that

supports MLI in general should provide appropriate mechanisms and metrics for the "transfer" of credentials.

22.3.8   Negotiation and Enforcement of Privacy Policies and Preferences

When entities interact with each other, different requirements towards the handling of each partner's data come up. Depending on the other parties involved in the interaction, an entity may have differing preferences in terms of which PII to disclose and which PII from others to collect, to execute, and to store. Policies are negotiated according to the entity's intentions with regard to other parties' preferences and legal regulations. Especially in MLI, negotiation processes may become complex since interests of several parties have to be regarded. Besides the negotiation of privacy policies and preferences, mechanisms guaranteeing enforcement of the negotiated issues are required.

Within PRIME, Data Handling Policies (DHPs) provide support for negotiation. However – as indicated in Section 22.3.2 about *relationship information* – they need to be extended in order to be applicable for negotiation between users. The current implementation of the CeL prototype does not integrate DHPs or other mechanisms for negotiation of privacy policies and preferences. From a community perspective, such a policy does neither touch interactions between users and content operators – e.g., learners and owners of workspaces – nor between users themselves while being involved in multiple collaborations. However, this should be possible for MLI environments.

22.3.9   Workflows and Behaviour Patterns

Each entity conducts individual workflows and behaviour patterns which could identify this entity and therefore represent PII. In order to protect privacy, the identity management system should offer possibilities to avoid those individual workflows and behaviour patterns, e.g. by providing templates. From a usability perspective, it would be necessary to still let users act according to their individual habits and to equip their client devices with functionalities for transferring the users' unstructured input into structured input for templates. Templates are either

- created by the designers of the system beforehand, or
- generated by the entities themselves and offered to other entities not only for altruistic, but also egoistic reasons, namely to increase privacy of the former.

In both cases, it needs to be ensured that entities can trust the templates, i.e., that there are no hidden channels, for instance.

The CeL prototype currently offers templates for workspace and content creation processes. Thus, the main intention is not on workflows and behaviour patterns, but on decisions about functionalities and layout of elements. Both

aspects could also be an indicator for personal preferences of users, thus the template approach could decrease options for identification in that context as well. Future work needs to identify and analyse typical workflows and behaviours in order to provide further templates covering workflows and behaviour patterns. Furthermore, investigations of applications with different levels of complexity are necessary, which allow one to identify useful templates decreasing patterns of personal behaviour without hindering the work within the application in general.

22.3.10   External Regulations

Legal requirements and regulations play an important role in all processes of personal data handling. They have to be considered alongside the interests of interaction partners when decisions about the handling of PII are made. Since legislations differ between regions or countries and may change over time, this component has to be regionally harmonised and needs to be kept up-to-date at all times. Currently in the year 2008, legal regulations regarding privacy mainly focus on interactions between state and citizen as well as between service provider and customer. A central problem is that mandatory regulations for relationships among individuals exist in the context of private law, only.

Within PRIME, besides technical aspects, legal requirements and given regulations have been analysed and influenced the design of the identity management components [WP608]. For instance, the so-called "Send Personal Data" dialogue considers user consent and control of use and storage of their PII in accordance with EU Directive 95/46/EC [Dir08], (cf. Chapters 20, 5). The user decides by herself whether to disclose PII for a certain purpose to specified recipients. Figure 22.2 shows the realisation of that dialogue within the CeL prototype.

Further, when first starting the eLearning system, a privacy policy is displayed which explains the handling of user data including PII by the central server – if such information is accumulated at all. This privacy policy from the provider of the technical platform is a static one in the current implementation and does neither inform a user of which of her data are stored on the server, nor does it allow for definition of rules regarding storage of her PII. Since user data are not stored on the server to a great extent, this mainly concerns data from owners of workspaces and data from partial identities that have edited some contents.

Additionally, users are able to store PII of partial identities of their interaction partners, e.g. contact data in their personal address books. The current implementation does not offer static or dynamic hints on regulations. Personal data such as address books and credentials are always stored locally on the client side within the PRIME middleware component of the CeL prototype.
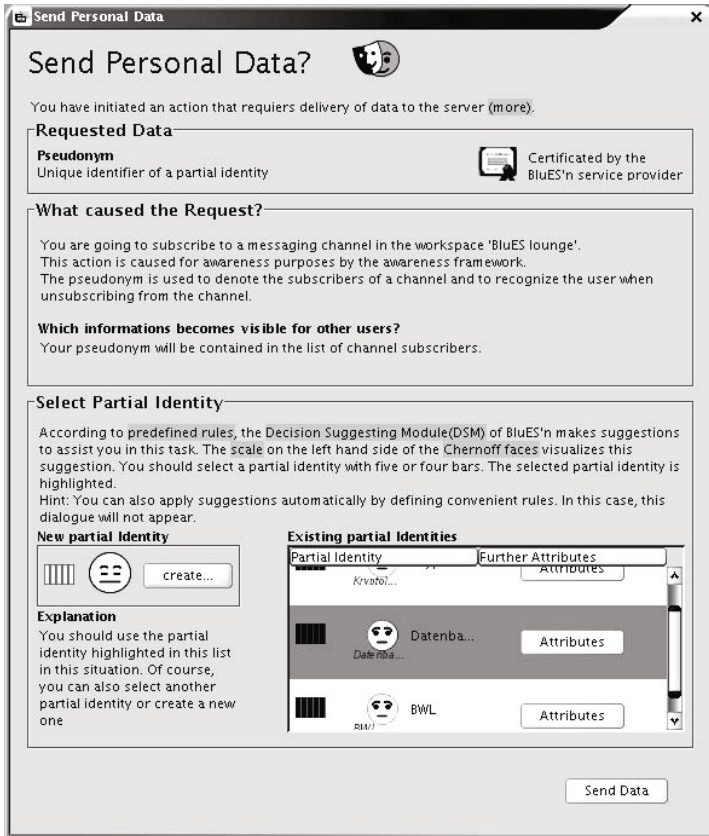
Fig. 22.2 Integration of legal data protection rules in the User Interface of the PRIME CeL prototype

22.4 Summary and Outlook

22.4.1 Overview of Building Blocks

Identity management systems for MLI that can be integrated, for instance, with a collaborative privacy-enhancing eLearning environment, need to take into account requirements from entities which act in various contexts and which maintain relationships with many other entities. With respect to the technical realisation of such identity management systems for MLI, also performance of such a system needs consideration, i.e. generating, managing, and selecting partial identities should be efficient and effective.

The introduced building blocks serve as functional components of such an identity management system. Some of these building blocks are realised and evaluated in the context of the project PRIME. Other building blocks

that evolve from and focus more on requirements from MLI have to be further investigated and developed in future research. Table 22.1 further below summarises all presented building blocks and gives an overview of realised components taken from PRIME and the CeL prototype as well as extended requirements.

22.4.2   Building Blocks in the Model of David Chaum

Finally, after the building blocks for extended requirements from MLI are presented, it could be asked how they fit into traditional user/service provider scenarios. In order to indicate similarities and extensions made to David Chaum's model which is introduced at the beginning of this chapter, we show the roles of the different building blocks for MLI in Chaum's model in the following.

Individuals in their role as *service users* in Chaum's model need building blocks for:

- **Partial Identities and Pseudonyms:** Individuals use digital pseudonyms in their interactions with organisations. This is a basic idea in Chaum's model.
- **Trust Management and Reputation:** Individuals only have relationships to organisations in their role as service providers. These organisations have no common identifiers for individuals and are not able to exchange individuals' PII. Thus, individuals do not need to worry about a service provider's trustworthiness regarding the handling of their PII. Still, reputation systems can support users in finding providers with good reviews for specific services.
- **Context and History:** This building block supports the individual in managing disclosure of PII to organisations including pseudonyms by referring to data from past interactions. However, assuming that organisations cannot exchange data on individuals, no further insight – e.g. estimation of linkability for partial identities – is to be expected.
- **Access Control:** Individuals need anonymous convertible credentials in order to get access to resources of organisations. Individuals do not generate credentials themselves, in order to protect their resources.
- **External regulations:** For individuals in their role as service users, there are mandatory regulations for interactions with service providers of interest. National differences and updates need to be considered. If, for instance, data retention policies in Germany make it necessary for providers of communication services to store all communication data of their users for a defined period, users should be informed about that issue before they decide to communicate.

Organisations in their role as *service providers* need the following building blocks in Chaum's model:

- **Trust Management and Reputation:** Organisations interact with individuals which use pseudonyms. In order to minimise damage caused by misrepresentation of individuals, reputations can serve as indicator. Chaum's extended model provides an automated means to help organisations to establish trust in the integrity of individuals' data by introducing tamper-resistant modules on the client side.
- **Access Control:** Organisations provide anonymous convertible credentials and protect their resources by granting access only if requested credentials are shown.
- **External Regulations:** For organisations in their role as service providers, there are mandatory regulations for their interactions with individuals of interest. National differences and updates need to be considered.

In the world of Chaum, relationships exist only between individuals and organisations, where the latter never deal pseudonymously or even anonymously. On the contrary, contact data of organisation by nature are always public. These organisations determine privacy policies and provide workflows in their role as service providers. Furthermore, since Chaum has no focus on collaboration, searching for and finding of interaction partners is less relevant in this context. The same is true for extended and privacy-enhancing awareness information which is of less importance in structured relationships between one individual and one organisation.

In the context of MLI, differences between service providers on the one hand and users on the other hand are no longer decisive and we use the more general term of entities. In Internet communities – e.g. auction platforms or collaborative eLearning environments – each entity may use services and provide services as well and collaborate with other entities. Based on these modified starting conditions which necessarily differ from those of Chaum's model due to the focus on MLI, there is a need for an extended identity management which contains such building blocks as they have been introduced earlier in Chapter (22.3).

22.4.3 Research Questions

A fundamental principle of user-controlled privacy emphasises that every single entity – lawyers speak of "data subjects" – has control over its personal information. If data processing is not dictated by law, this entity decides if, where, for what purpose and how long its personal information is stored [Dir08]. With regard to current technical development, governmental institutions and organisations are no longer the only entities being able to generate, process, and store PII to a great extent. In contrast, also individuals are supported. Within the scope of multilateral interactions, the current understanding of privacy is not sufficient. It needs to be scrutinised and extended since relationship information is not only assigned to one, but to two or more entities and represents personal information of all entities involved in that relation. Additionally, it is to be expected that future data protection laws will

have a stronger focus on relationships between individuals on top of the existing regulations for relationships between *state and citizens*, and *organisations and customers.*

In this scope, it needs to be discussed further whether it is legally reasonable for legal persons to have a right to protection of their PII similar to natural persons, as it is established in the Data Protection Act of Austria [Dat08].

Open research questions in the field of MLI that concentrate especially on *relationship information* are for instance:

- How does relationship information need to be handled in a privacy-enhancing way while respecting multiple privacy interests as well?
- How could a negotiation process for the storage and disclosure of relation information be designed and automated?
- Does prohibiting storage of relation information affect user control of a single entity?
- How does relation information change over time?

With regard to all building blocks, more open issues and questions arise. It needs to be investigated how presentation and use of awareness information that support collaboration can be designed and realised in a privacy-respecting manner, for instance. Another aspect that requires further research are solutions for user-specific definitions of policies in order to support negotiation processes between arbitrary entities.

Table 22.1 Overview of Building Blocks, realised components for PRIME Middleware or CeL prototype and extended requirements from an MLI perspective

Building Block	PRIME-Middleware or CeL prototype provide	Extended Requirements from MLI
Pseudonyms and Partial Identities	• partial identities and pseudonyms as identifiers for users	• confidential communication within a dynamic subgroup of entities or partial identities, respectively
Relationship Information	• overview of contacts (Data Track) • list of friends (CeL)	• incorporation of privacy interests of all relationship partners (storage, representation, e.g., in map(s) of social networks)

Table 22.2 (*continued*)

Building Block	PRIME-Middleware or CeL prototype provide	Extended Requirements from MLI
Searching for and Finding of Interaction Partners	• contact data known by the public (service providers) • past interaction partners in Data Track overview	• intentional linkage of partial identities possible for their holder
Trust Management and Reputation	• Platform Trust Manager	• privacy-respecting, platform independent reputation system, adaptable by each user relying on reputation
Awareness Information	• information from the application that contributes to Workspace Awareness (CeL, PRIME) • privacy-enhancing awareness information that contributes to Privacy Awareness (CeL)	• privacy-enhancing awareness information for MLI scenarios (Privacy Awareness) • collaboration-supporting awareness information (Group Awareness)
Context and History	• Data Track • Context Management (CeL)	• general Context Management for various applications
Access Control	• anonymous convertible credentials	• anonymous convertible credentials for use in any application
Negotiation and Enforcement of Privacy Policies and Preferences	• Data Handling Policies between client and server	• general Data Handling Policies • user-specified policies

Table 22.3 (*continued*)

Building Block	PRIME-Middleware or CeL prototype provide	Extended Requirements from MLI
Workflows and Behaviour Patterns	• workspace templates, content templates, layout templates (CeL)	• workflow templates
External Regulations	• consideration of legal requirements for design and implementation	• more flexible integration of external guidelines (national differences, differences ensuing from entities, updates of regulations)

References

[ACC+05]   Andersson, C., Camenisch, J., Crane, S., Fischer-Hübner, S., Leenes,
            R., Pearson, S., Pettersson, J.S., Sommer, D.: Trust in Prime. In:
            Proceedings of the 5th IEEE Int. Symposium on Signal Processing
            and IT, Athens, Greece, December 18-21 (2005)

[ACD+06]   Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimer-
            cati, S., Samarati, P.: Supporting location-based conditions in access
            control policies. In: Proc. of the ACM Symposium on Information,
            Computer and Communications Security (ASIACCS 2006), Taipei,
            Taiwan (March 2006)

[ACDS08]   Ardagna, C.A., Cremonini, M., De Capitani di Vimercati, S., Sama-
            rati, P.: A privacy-aware access control system. Journal of Computer
            Security (JCS) 16(4), 369–392 (2008)

[ACJT00]   Ateniese, G., Camenisch, J.L., Joye, M., Tsudik, G.: A practical
            and provably secure coalition-resistant group signature scheme. In:
            Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270.
            Springer, Heidelberg (2000)

[Ack04]   Ackerman, M.S.: Privacy in pervasive environments: next generation
            labelling protocols. In: Personal Ubiquitous Computing 2004, pp.
            430–439. Springer, Heidelberg (2004)

[ACM00]   ACM00. Special issue on "trusting technology". Communications of
            the ACM 43(12) (2000)

[Acq04]   Acquisti, A.: Privacy in electronic commerce and the economics of
            immediate gratification. In: EC 2004: Proceedings of the 5th ACM
            Conference on Electronic Commerce, pp. 21–29. ACM, New York
            (2004)

[Act03]   Gramm-Leach-Bliley Act. Gramm-leach-bliley act: Privacy of con-
            sumer financial information (2003),
            http://www.ftc.gov/privacy/glbact/glboutline.htm

[ADD+05]   Ardagna, C.A., Damiani, E., De Capitani di Vimercati, S., Fugazza,
            C., Samarati, P.: Offline Expansion of XACML Policies Based on
            P3P Metadata. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005.
            LNCS, vol. 3579, pp. 363–374. Springer, Heidelberg (2005)

[ADDS05]   Ardagna, C.A., Damiani, E., De Capitani di Vimercati, S., Samarati,
            P.: Towards privacy-enhanced authorization policies and languages.
            In: Jajodia, S., Wijesekera, D. (eds.) Data and Applications Security
            2005. LNCS, vol. 3654, pp. 16–27. Springer, Heidelberg (2005)
[ADS06]   Ardagna, C.A., De Capitani di Vimercati, S., Samarati, P.: Enhanc-
            ing user privacy through data handling policies. In: Damiani, E., Liu,
            P. (eds.) Data and Applications Security 2006. LNCS, vol. 4127, pp.
            224–236. Springer, Heidelberg (2006)
[AG06]   Anwar, M., Greer, J.: Reputation management in privacy-enhanced
            e-learning. In: I2LOR 2006 Proceedings (November 2006)
[AG07]   Adelsbach, A., Greveler, U.: Insider attacks enabling data broad-
            casting on crypto-enforced unicast links. In: Biskup, J., López, J.
            (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 469–484. Springer, Hei-
            delberg (2007)
[AHK$^+$03] Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.:
            Enterprise privacy authorization language, epal 1.1 (2003),
            http://www.zurich.ibm.com/security/enterprise-privacy/epal
[AHKS02]   Ashley, P., Hada, S., Karjoth, G., Schunter, M.: E-P3P privacy poli-
            cies and privacy authorization. In: Proc. of the ACM Workshop on
            Privacy in the Electronic Society (WPES 2002), Washington, DC,
            USA (November 2002)
[AIR01]   Aiello, W., Ishai, Y., Reingold, O.: Priced oblivious transfer: How
            to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001.
            LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001)
[AMG05]   Aguilar-Melchor, C., Gaborit, P.: Lattice-based computationally-
            efficient private information retrieval protocol. In: Western Euro-
            pean Workshop on Research in Cryptology (2005)
[Ars]   Arshad, F.: Privacy fox: A javascript-based p3p agent for mozilla
            firefox, http://privacyfox.mozdev.org/PaperFinal.pdf
[Art04]   Article 29 Data Protection Working Party. Opinion on More Har-
            monised Information Provisions (November 25, 2004),
            http://www.europa.eu.int/comm/justice_home/fsj/privacy/
            docs/wpdocs/2004/wp100_en.pdf
[AT&]   AT&T Privacy Bird, http://www.privacybird.com
[Ate99]   Ateniese, G.: Efficient verifiable encryption (and fair exchange) of
            digital signatures. In: Proc. 6th ACM Conference on Computer
            and Communications Security, pp. 138–146. ACM Press, New York
            (1999)
[Bao00]   Bao, F.: An efficient verifiable encryption scheme for the encryp-
            tion of discrete logarithms. In: Schneier, B., Quisquater, J.-J. (eds.)
            CARDIS 1998. LNCS, vol. 1820. Springer, Heidelberg (2000)
[BCC04]   Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation,
            pp. 225–234. ACM Press, New York (2004)
[BCL04]   Bangerter, E., Camenisch, J., Lysyanskaya, A.: A cryptographic
            framework for the controlled release of certified data. In: Christian-
            son, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols
            2004. LNCS, vol. 3957, pp. 20–42. Springer, Heidelberg (2006)

[BCM05]   Bangerter, E., Camenisch, J., Maurer, U.M.: Efficient proofs of
            knowledge of discrete logarithms and representations in groups with
            hidden order. In: Hutter, D., Ullmann, M. (eds.) PKC 2005. LNCS,
            vol. 3386, pp. 154–171. Springer, Heidelberg (2005)

[BCP⁺02]   Balacheff, B., Chen, L., Pearson, S., Plaquin, D., Proudler, G.:
            Trusted Computing Platforms. Prentice Hall, Englewood Cliffs
            (2002)

[BCR87]   Brassard, G., Crépeau, C., Robert, J.M.: All-or-nothing disclosure
            of secrets. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263,
            pp. 234–238. Springer, Heidelberg (1987)

[BdB90]   Bos, J.N., den Boer, B.: Detection of Disrupters in the DC Proto-
            col. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989.
            LNCS, vol. 434, pp. 320–327. Springer, Heidelberg (1990)

[BDDS01]   Bonatti, P., Damiani, E., De Capitani di Vimercati, S., Samarati,
            P.: A component-based architecture for secure data publication. In:
            Proc. of the 17th Annual Computer Security Applications Confer-
            ence (ACSAC 2001), New Orleans, LA (December 2001)

[Ber05]   Bergmann, M.: Integrated Prototype V1. PRIME Deliverable 11.2.a
            (September 2005)

[Ber07]   Bergmann, M.: User-side IDM Integrated Prototype V2 adjunctive
            documentation. PRIME Deliverable 11.2.b (March 2007)

[Ber08]   Bergmann, M.: Generic Predefined Privacy Preferences for Online
            Applications. In: Hübner, S.F., Duquenoy, P., Zuccato, A., Mar-
            tucci, L. (eds.) The Future of Identity in the Information Society:
            Proceedings of the Third IFIP WG 9.2, 9.6/11.6, 11.7/FIDIS In-
            ternational Summer School, International Summerschool Karlstad,
            Sweden, May 15. Springer, Heidelberg (2008),
            http://www.cs.kau.se/IFIP-summerschool/papers/
            S05_P4_Mike_Bergmann.pdf

[BFIK98]   Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D.: The role
            of trust management in distributed systems security. In: Secure In-
            ternet Programming: Issues in Distributed and Mobile Object Sys-
            tems (1998)

[BFL96]   Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust manage-
            ment. In: Proc. of 1996 IEEE Symposium on Security and Privacy,
            Oakland, CA, USA, pp. 164–173 (May 1996)

[BG03]   Bennett, K., Grothoff, C.: GAP – Practical Anonymous Networking.
            In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 141–160.
            Springer, Heidelberg (2003)

[Bic07]   Bichsel, P.: Theft and misuse protection for anonymous credentials.
            Techn. report, ETH Zürich (November 2007),
            ftp://ftp.tik.ee.ethz.ch/pub/students/
            2007HS/MA200742.pdf

[BIK03]   Blaze, M., Ioannidis, J., Keromytis, A.D.: Experience with the
            keyNote trust management system: Applications and future direc-
            tions. In: Nixon, P., Terzis, S. (eds.) iTrust 2003. LNCS, vol. 2692,
            pp. 284–300. Springer, Heidelberg (2003)

[BJWW02]   Bettini, C., Jajodia, S., Sean Wang, X., Wijesekera, D.: Provisions and obligations in policy management and security applications. In: Proc. 28th Conference Very Large Data Bases (VLDB 2002), Hong Kong, China (August 2002)

[BK06]   Barker, E., Kelsey, J.: Recommendation for random number generation using deterministic random bit generators. SP-800-90, U.S. DoC/National Institute of Standards and Technology (2006), http://csrc.nist.gov/publications/nistpubs/

[BKP90]   Bethlehem, J.G., Keller, W.J., Pannekoek, J.: Disclosure control of microdata. Journal of The American Statistical Association 85, 38–45 (1990)

[Blo97]   Blobel, B.: Personal medical information: Security, engineering and ethics. In: Information Security Solutions Europe Conference (ISSE), pp. 39–56. R.J. Anderson (1997)

[BPHL+06a]   Borcea-Pfitzmann, K., Hansen, M., Liesebach, K., Pfitzmann, A., Steinbrecher, S.: What user-controlled identity management should learn from communities. Information Security Technical Report 11(3), 119–128 (2006)

[BPHL+06b]   Borcea-Pfitzmann, K., Hansen, M., Liesebach, K., Pfitzmann, A., Steinbrecher, S.: What User-Controlled Identity Management Should Learn from Communities. Information Security Technical Report 11(3), 119–128 (2006)

[BPHL+07a]   Borcea-Pfitzmann, K., Hansen, M., Liesebach, K., Pfitzmann, A., Steinbrecher, S.: Managing one's identities in organisational and social settings. DuD, Datenschutz und Datensicherheit 31(9), 671–675 (2007)

[BPHL+07b]   Borcea-Pfitzmann, K., Hansen, M., Liesebach, K., Pfitzmann, A., Steinbrecher, S.: Managing One's Identities in Organisational and Social Settings. Datenschutz und Datensicherheit 9 (2007)

[BPL07]   Borcea-Pfitzmann, K., Liesebach, K.: The concept of workspaces – re-defined for eLearning. Advanced Technology for Learning (ATL) International Journal, Special Issue on Technology for Collaborative Learning 4(1) (2007)

[BR01]   Buskens, V., Raub, W.: Embedded trust: Control and learning. In: Lawler, E., Thye, S. (eds.) Group Cohesion, Trust, and Solidarity. Advances in Group Processes, vol. 19, pp. 167–202 (2001)

[Bra93]   Brands, S.: Untraceable Off-Line Cash in Wallets with Observers. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (1994)

[Bra95]   Brands, S.: Restrictive blinding of secret-key certificates. Technical Report CS-R9509, CWI (September 1995)

[Bra97]   Brands, S.: Rapid Demonstration of Linear Relations Connected by Boolean Operators. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 318–333. Springer, Heidelberg (1997)

[Bra99]   Brands, S.: Rethinking Public Key Infrastructure and Digital Certificates— Building in Privacy. PhD thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands (1999)

[Bra00]   Brands, S.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)

[BRP05]   Bergmann, M., Rost, M., Pettersson, J.S.: Exploring the Feasibility of a Spatial User Interface Paradigm for Privacy-Enhancing Technology. In: Proceedings of the Fourteenth International Conference on Information Systems Development, Karlstad. Springer, Heidelberg (2005)

[BS02a]   Bonatti, P., Samarati, P.: A unified framework for regulating access and information release on the web. Journal of Computer Security 10(3), 241–272 (2002)

[BS02b]   Bonatti, P.A., Samarati, P.: A uniform framework for regulating service access and information release on the web. Journal of Computer Security 10(3), 241–272 (2002)

[BSCGS06]   Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D.: User centricity: A taxonomy and open issues. In: ACM Digital Identity Management 2006 (DIM 2006). ACM Press, New York (2006)

[Byg02]   Bygrave, L.: Data Protection Law, Approaching Its Rationale, Logic and Limits. Kluwer Law International, The Hague (2002)

[CAG02]   Cranor, L.F., Arjula, M., Guduru, P.: Use of a P3P User Agent by Early Adopters. In: Proceedings of the ACM Workshop on Privacy in the Electronic Society, November 21 (2002)

[Cas04a]   Casassa Mont, M.: Ddealing with privacy obligations in enterprises. In: Securing Electronic Businss Processes, ISSE 2004, Berlin, Germany (September 2004)

[Cas04b]   Casassa Mont, M.: Dealing with privacy obligations: Important aspects and technical approaches. In: Katsikas, S.K., López, J., Pernul, G. (eds.) TrustBus 2004. LNCS, vol. 3184, pp. 120–131. Springer, Heidelberg (2004)

[Cas04c]   Casassa Mont, M.: A system to handle privacy obligations in enterprises, hpl-2005-180

[Cas06]   Casassa Mont, M.: Towards scalable management of privacy obligations in enterprises. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 1–10. Springer, Heidelberg (2006)

[CasSZ06]   Camenisch, J., Shelat, A., Sommer, D., Zimmermann, R.: Securing user inputs for the web. In: Conference on Computer and Communications Security, Proceedings of the Second ACM Workshop on Digital Identity Management, pp. 33–44 (2006)

[CB95]   Cooper, D.A., Birman, K.: Preserving Privacy in a Network of Mobile computers. In: 1995 IEEE Symposium on Security and Privacy (S&P 1995), Oakland, CA, USA, pp. 26–38. IEEE Computer Society Press, Los Alamitos (1995)

[CB07a]   Casassa Mont, M., Beato, F.: On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises. In: Proc. of the 8th IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2007), Bologna, Italy (June 2007)

[CB07b]   Casassa Mont, M., Beato, F.: On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises. In: 8th IEEE Conference on Policies for Distributed Systems and Networks (Policy 2007), Bologna, Italy (June 2007)

[CC00]   Cavoukian, A., Crompton, M.: Web seals: A review of online privacy programs. In: Proc. 22nd International Conference on Privacy and Data Protection (2000),
http://www.privacy.gov.au/publications/seals.pdf

[CCS06]   Casassa Mont, M., Crosta, S., Sommer, D.: PRIME Architecture V2. Technical report, PRIME Project (2006),
https://www.prime-project.eu/prime_products/reports/arch/

[CD00]   Camenisch, J., Damgård, I.: Verifiable Encryption, Group Encryption, and Their Applications to Separable Group Signatures and Signature Sharing Schemes. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 331–345. Springer, Heidelberg (2000)

[CE87]   Chaum, D., Evertse, J.-H.: A Secure and Privacy-Protecting Protocol for Transmitting Personal Information between Organizations. In: Odlyzko, M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 118–167. Springer, Heidelberg (1987)

[CFL+97]   Chu, Y.-H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: Referee: Trust management for web applications. Computer Networks and ISDN Systems 29(8-13), 953–964 (1997)

[CFN90]   Chaum, D., Fiat, A., Naor, M.: Untraceable Electronic Cash. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 319–327. Springer, Heidelberg (1990)

[CG05]   Cranor, L.F., Garfinkel, S.: Security and Usability: Designing Secure Systems that People Can Use. O'Reilly, Sebastopol (2005)

[CGKS95]   Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: FOCS, pp. 41–50 (1995)

[CGP+08]   Cimato, S., Gamassi, M., Piuri, V., Sassi, R., Scotti, F.: Privacy-aware biometrics: Design and implementation of a multimodal verification system. In: Proc. of the Annual Computer Security Applications Conference (ACSAC 2008), Anaheim, CA, USA (December 2008)

[Cha81]   Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24(2), 84–88 (1981)

[Cha85a]   Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM 28(10), 1030–1044 (1985)

[Cha85b]   Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM 28(10) (1985)

[Cha86]   Chaum, D.: Showing credentials without identification. Signatures transferred between unconditionally unlinkable pseudonyms. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 241–244. Springer, Heidelberg (1986)

[Cha88]   Chaum, D.L.: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Journal of Cryptology (1), 65–75 (1988)

[Cha92]   Chaum, D.: Achieving Electronic Privacy. Scientific American 2(276), 96–101 (1992)

[Cha04]   Chang, Y.-C.: Single Database Private Information Retrieval with
            Logarithmic Communication. In: Wang, H., Pieprzyk, J., Varad-
            harajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 50–61.
            Springer, Heidelberg (2004)
[Che99]   Cheskin. Research and studio archetype (January 1999),
            http://www.sapient.com/cheskin/
[Che00]   Chen, Z.: Java Card Technology for Smart Cards: Architecture and
            Programmer's Guide. Addison-Wesley, Reading (2000)
[CHK+06]   Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A.,
            Meyerovich, M.: How to win the clonewars: efficient periodic n-times
            anonymous authentication. In: ACM Conference on Computer and
            Communications Security 2006. ACM Press, New York (2006)
[CK03]   Clauß,   S.,   Kriegelstein,   T.:   Datenschutzfreundliches   Iden-
            titätsmanagement. DuD Datenschutz und Datensicherheit 27,
            297 (2003)
[CKGS98]   Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private infor-
            mation retrieval. J. ACM 45(6), 965–981 (1998)
[CL00]   Camenisch, J., Lysyanskaya, A.: Efficient non-transferable anony-
            mous multi-show credential system with optional anonymity revo-
            cation. Technical Report Research Report RZ 3295, IBM Research
            Division (November 2000)
[CL01a]   Camenisch, J., Lysyanskaya, A.: Efficient non-transferable anony-
            mous multi-show credential system with optional anonymity revoca-
            tion. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045,
            pp. 93–118. Springer, Heidelberg (2001)
[CL01b]   Camenisch, J., Lysyanskaya, A.: Efficient non-transferable anony-
            mous multi-show credential system with optional anonymity revoca-
            tion. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045,
            pp. 93–118. Springer, Heidelberg (2001)
[CL01c]   Camenisch, J., Lysyanskaya, A.: An identity escrow scheme with
            appointed verifiers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS,
            vol. 2139, pp. 388–407. Springer, Heidelberg (2001)
[CL02]   Camenisch, J., Lysyanskaya, A.: Dynamic Accumulators and Appli-
            cation to Efficient Revocation of Anonymous Credentials. In: Yung,
            M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Hei-
            delberg (2002)
[CL03]   Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient
            Protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002.
            LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)
[CL04]   Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous
            credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO
            2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
[Claa]   Clarke, R.: P3p re-visited,
            http://www.anu.edu.au/people/Roger.Clarke/DV/P3PRev.html
[Clab]   Clarke, R.: Platform for privacy preferences: A critique,
            http://www.anu.edu.au/people/Roger.Clarke/DV/P3PCrit.html
[Cla06]   Clauß, S.: A framework for quantification of linkability within a
            privacy-enhancing identity management system. In: Müller, G. (ed.)
            ETRICS 2006. LNCS, vol. 3995, pp. 191–205. Springer, Heidelberg
            (2006)

634   References

[Cla07a]   Clauß, S.: Towards Quantification of Privacy Within a Privacy–
            Enhancing Identity Management System. PhD thesis, Technische
            Universität Dresden, Saxony, Germany (2007)
[Cla07b]   Clauß, S.: Towards Quantification of Privacy Within a Privacy-
            Enhancing Identity Management System. PhD thesis, Technische
            Universität Dresden (December 2007)
[CMO00]   Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single database pri-
            vate information retrieval implies oblivious transfer. In: Preneel, B.
            (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 122–138. Springer,
            Heidelberg (2000)
[CMS99]   Cachin, C., Micali, S., Stadler, M.: Computationally Private Infor-
            mation Retrieval with Polylogarithmic Communication. In: Stern, J.
            (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer,
            Heidelberg (1999)
[CNS07]   Camenisch, J.L., Neven, G., Shelat, A.: Simulatable Adaptive Obliv-
            ious Transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS,
            vol. 4515, pp. 573–590. Springer, Heidelberg (2007)
[Com03]   UK Information Commissioner. Who cares about data protec-
            tion? (2003), http://www.informationcommissioner.gov.uk/cms/
            DocumentUploads/Segmentation%20Research%20Findings.pdf
[Con08]   Open Trusted Computing Consortium (2008),
            http://www.opentc.net/
[Cou95]   Council of Europe. Data Protection Directive 1995/46/EC of the
            European Parliament and of the Council of 24 October 1995, on the
            protection of individuals with regard to the processing of personal
            data and on the free movement of such data, 1995. Official Journal
            L No. 281 (November 23, 1995)
[CPB03]   Casassa Mont, M., Pearson, S., Bramhall, P.: Towards accountable
            management of privacy and identity information. In: Snekkenes, E.,
            Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 146–161.
            Springer, Heidelberg (2003)
[Cra02]   Cranor, L.F.: Web Privacy with P3P. O'Reilly & Associates, Se-
            bastopol (2002)
[Cra06]   Cranor, L.F.: What do they "indicate?": evaluating security and
            privacy indicators. Interactions XIII.3, 45–57 (2006)
[CS97a]   Camenisch, J., Stadler, M.: Efficient Group Signature Schemes for
            Large Groups. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS,
            vol. 1294, pp. 410–424. Springer, Heidelberg (1997)
[CS97b]   Camenisch, J., Stadler, M.: Proof systems for general statements
            about discrete logarithms. Technical Report TR 260, Institute for
            Theoretical Computer Science, ETH Zürich (March 1997)
[CS98]   Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem
            Provably Secure against Adaptive Chosen Ciphertext Attack. In:
            Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25.
            Springer, Heidelberg (1998)
[CS03]   Camenisch, J., Shoup, V.: Practical Verifiable Encryption and De-
            cryption of Discrete Logarithms. In: Boneh, D. (ed.) CRYPTO 2003.
            LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)

[CS06]   Clauß, S., Schiffner, S.: Structuring anonymity metrics. In: Goto, A. (ed.) Proceedings of the Second ACM Workshop on Digital Identity Management, Alexandria, Virginia, USA, pp. 55–62. ACM, New York (2006)

[CSSZ06]   Camenisch, J., Shelat, A., Sommer, D., Zimmermann, R.: Securing user inputs for the web. In: ACM Digital Identity Management 2006 (DIM 2006). ACM Press, New York (2006)

[CSZ06]   Camenisch, J., Sommer, D., Zimmermann, R.: A general certification framework with applications to privacy-enhancing certificate infrastructures. In: Proceedings of SEC 2006. Springer, Heidelberg (2006)

[CT01]   Castelfranchi, C., Tan, Y.-H. (eds.): Trust and Deception in Virtual Societies. Kluwer Academic Publishers, Dordrecht (2001)

[CT05]   Chu, C.-K., Tzeng, W.-G.: Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 172–183. Springer, Heidelberg (2005)

[CV02]   Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: Proc. of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washingtion, DC, USA, November 2002, pp. 21–30 (2002)

[CvH91]   Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)

[Dam90]   Damgård, I.B.: Payment Systems and Credential Mechanisms with Provable Security against Abuse by Individuals. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 328–335. Springer, Heidelberg (1990)

[Dan03]   Danezis, G.: Better Anonymous Communications. PhD thesis, University of Cambridge (December 2003)

[Dat08]   Datenschutzgesetz 2000 (DSG 2000), BGBl. I Nr. 165/1999 (March 2008), http://www.dsk.gv.at/dsg2000d.htm

[DD08]   Dhamija, R., Dusseault, L.: The Seven Flaws of Identity Management - Usability and Security Challenges. IEEE Security and Privacy 6(2), 22–29 (2008)

[DDFS04]   Damiani, E., De Capitani di Vimercati, S., Fugazza, C., Samarati, P.: Extending policy languages to the semantic web. In: Koch, N., Fraternali, P., Wirsing, M. (eds.) ICWE 2004. LNCS, vol. 3140, pp. 330–343. Springer, Heidelberg (2004)

[Del00]   Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: EC 2000: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 150–157. ACM Press, New York (2000)

[Del03]   Dellarocas, C.: The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms. Management Science (2003)

[Del06]   Dellarocas, C.: Research note – how often should reputation mechanisms update a trader's reputation profile? Information Systems Research 17(3), 271–285 (2006)

[Dir95]   Directive 95/46/ec of the european parliament and of the council of 24 October 1995, on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal L 281, 31–50 (November 23, 1995)

[Dir08]   Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (March 2008), `europa.eu.int/eurlex/lex/LexUriServ/LexUriServ.dou?ri=CELEX:31995L0046:EN:HTML`

[DJ01]   Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)

[DMS04]   Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (2004)

[DO00]   Dolev, S., Ostrovsky, R.: Xor-trees for efficient anonymous multicast and reception. ACM Trans. Inf. Syst. Secur. 3(2), 63–84 (2000)

[Dou02]   Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)

[DRC98]   Burt, R., Rousseau, D., Sitkin, S., Camerer, C.: Not so different after all: a cross-discipline view of trust. Academy of Management Review 23(3), 393–404 (1998)

[DSCP02]   Díaz, C., Seys, S., Claessens, J., Preneel, B.: Towards Measuring Anonymity. In: Syverson, P., Dingledine, R. (eds.) PET 2002. LNCS, vol. 2482, pp. 54–68. Springer, Heidelberg (2003)

[DTH06]   Dhamija, R., Tygar, J.D., Hearst, M.: Why Phishing Works. In: CHI 2006 Proceedings. ACM Conference Proceedings Series. ACM Press, New York (2006)

[Egg98]   Egger, F.N.: Increasing consumers' confidence in electronic commerce through human factors engineering (1998)

[EGL85]   Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Communications of the Association for Computing Machinery 28(6), 637–647 (1985)

[Eri02]   Erickson, J.: Fair use, drm and trusted computing: Friends or foes? Comm. of ACM (2002)

[Esk01]   Eskor, T.: Quality Assurance in Corporate IT: It Matters More than Ever. Technical report, Los Angeles, USA (2001)

[EU006]   European Union Internet Action Plan: Filtering & Rating, `http://europa.eu.int/information_society/activities/sip/projects/filtering/` (visited March 2006)

[Eur95]   European Parliament. Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal L, pp. 31–50 (1995)

[Eur03]   Eurobarometer. Eurobarometer (2003), `http://ec.europa.eu/public_opinion/index_en.htm` especially: `http://ec.europa.eu/public_opinion/archives/ebs/ebs_196_data_protection.pdf`

[Eur08]   Eurobarometer. Eurobarometer, data controllers' perceptions (no. 226) and citizens' perceptions (no. 225) (2008), http://ec.europa.eu/public_opinion/flash/fl_225_en.pdf, http://ec.europa.eu/public_opinion/flash/fl_226_en.pdf; http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/08/249&format=HTML&aged=0&language=EN&guiLanguage=en

[Exl04]   Exler, R.: Security and the Application Development Process (2004)

[eXt05]   eXtensible Access Control Markup Language (XACML) Version 2.0 (February 2005), http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[FE06]   Franz, E., Engel, B.: A Realization of Context Management Facilitating the Usage of Partial Identities. In: Proceedings of Workshop on Privacy-Enhanced Personalization, PEP 2006 (2006), http://www.isr.uci.edu/pep06

[Fed05]   Federrath, H.: Privacy Enhanced Technologies: Methods – Markets – Misuse. In: Katsikas, S.K., López, J., Pernul, G. (eds.) TrustBus 2005. LNCS, vol. 3592, pp. 1–9. Springer, Heidelberg (2005)

[Fei06]   Feig, C.: Konzeption und Realisierung eines Awareness-Frameworks für die kollaborative Lernumgebung BluES. Diploma thesis, Technische Universität Dresden (2006)

[FH87]   Fischer-Hübner, S.: Zur reidentifikationssicheren statistischen Auswertung personenbezogener Daten in staatlichen Datenbanken. Diplomarbeit, Universität Hamburg (December 1987) (in German)

[FH01]   Fischer-Hübner, S. (ed.): It-security and privacy: Design and use of privacy-enhancing security mechanisms. LNCS, vol. 1958, p. 35. Springer, Heidelberg (2001)

[FHP04]   Fischer-Hübner, S., Pettersson, J.S.: Evaluation of Early Prototypes. PRIME deliverable D6.1.b, December 1 (2004), https://www.prime-project.eu/prime_products/reports/eval/

[FHPB+07]   Fischer-Hübner, S., Pettersson, J.S., Bergmann, M., Hansen, M., Pearson, S., Casassa-Mont, M.: Digital Privacy. In: Acquisti, et al. (eds.) Digital Privacy – Theory, Technologies, and Practices, Auerbach Publications (2007)

[FJK06]   Furnell, S.M., Jusoh, A., Katsabas, D.: The challenges of understanding and using security: A survey of end-users. Computers & Security 25(1), 27–35 (2006)

[FKH00]   Friedman, B., Kahn Jr., P.H., Howe, D.C.: Trust online. Communications of the ACM 43(12), 34–40 (2000)

[FM02]   Freedman, M.J., Morris, R.: Tarzan: A peer-to-peer anonymizing network layer. In: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington, DC (November 2002)

[For06]   FortConsult. Practical onion hacking: Finding the real address of tor clients (2006)

[FR99]   Friedman, E., Resnick, P.: The social cost of cheap pseudonyms. Journal of Economics and Management Strategy 10, 173–199 (1999)

[FS86]   Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)

[fSI07a]   International Organization for Standardization (ISO). Common Methodology for Information Technology Security Evaluation, Version 2.3, simular to IS 18405 - Evaluation Methodology Version 2.3. Technical report (2007)

[fSI07b]   International Organization for Standardization (ISO). The Common Criteria Part 3 - Version 2.3. Technical report (2007)

[Fur04a]   Furnell, S.M.: Using security: easier said than done? Computer Fraud & Security, pp. 6–16 (April 2004)

[Fur04b]   Furnell, S.M.: E-commerce Security: a Question of Trust. In: Computer Fraud & Securityt, pp. 10–14 (October 2004)

[Fur05]   Furnell, S.M.: Why users cannot use Security. Computers& Security 24(4), 274–279 (2005)

[GEB]   Gladman, B., Ellison, C., Bohm, N.: Digital signatures, certificates and electronic commerce, http://www.clark.net/pub/cme/html/spki.html

[Ger02]   Markotten, D.G.T.: User-centered security engineering. In: Proceedings of the 4th Europen/USENIX Conference NordU (February 2002), http://www.iig.uni-freiburg.de/telematik/atus/publications.html

[GGR96]   Gutwin, C., Greenberg, S., Roseman, M.: Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In: Proceedings of the Conference on Human-Computer Interaction, pp. 281–298 (1996)

[GHJV95]   Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: elements of reusable object-oriented software. Addison-Wesley Professional, Reading (1995)

[GHNR04]   Gunaratne, J., Hwong, B., Nelson, C., Rudorfer, A.: Using evolutionary prototypes to formalize product requirements. In: 'Bridging the Gaps II: Bridging the Gaps Between Software Engineering and Human-Computer Interaction' W1L Workshop - 26th International Conference of Software Engineering, Edinburgh, Scotland, UK, pp. 17–20 (2004)

[Gif00]   Giff, S.: The influence of metaphor, smart cards and interface dialogue on trust in ecommerce (2000)

[GJ04]   Golle, P., Juels, A.: Dining cryptographers revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 456–473. Springer, Heidelberg (2004)

[GLM⁺04]   Gamassi, M., Lazzaroni, M., Misino, M., Piuri, V., Sana, D., Scotti, F.: Accuracy and performance of biometric systems. In: Proc. of the 21st IEEE Instrumentation and Measurement Technology Conference (IMTC 2004), Como, Italy (May 2004)

[GMR88]   Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)

[GNO⁺04]   Gavriloaie, R., Nejdl, W., Olmedilla, D., Seamons, K.E., Winslett, M.: No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 342–356. Springer, Heidelberg (2004)

[Gol04]   Goldreich, O.: Foundations of Cryptography II: Basic Applications. Cambridge University Press, Cambridge (2004)

[GPSS05]   Gamassi, M., Piuri, V., Sana, D., Scotti, F.: Robust fingerprint detection for access control. In: Proc. of RoboCare Workshop 2005, Rome, Italy (May 2005)

[GR05]   Gentry, C., Ramzan, Z.: Single-Database Private Information Retrieval with Constant Communication Rate. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 803–815. Springer, Heidelberg (2005)

[Gre92]   Gregory, F.: SOFTWARE ASSURANCE STANDARD NASA-STD-2201-93. Washington D.C. (1992)

[GS03]   Grandison, T., Sloman, M.: Trust management tools for internet applications. In: Nixon, P., Terzis, S. (eds.) iTrust 2003. LNCS, vol. 2692, pp. 91–107. Springer, Heidelberg (2003)

[GS05]   Günther, O., Spiekermann, S.: RFID and the perception of control: The consumer's view. Communications of the ACM 48(9), 73–76 (2005)

[Hin02]   Hintz, A.: Fingerprinting websites using traffic analysis. In: Dingledine, R., Syverson, P. (eds.) PET 2002. LNCS, vol. 2482, pp. 171–178. Springer, Heidelberg (2003)

[HJW02]   Hogben, G., Jackson, T., Wilikens, M.: A fully compliant research implementation of the P3P standard for privacy protection: Experiences and recommendations. In: Gollmann, D., Karjoth, G., Waidner, M. (eds.) ESORICS 2002. LNCS, vol. 2502, pp. 104–125. Springer, Heidelberg (2002)

[HO03]   Halpern, J.Y., O'Neill, K.R.: Anonymity and information hiding in multiagent systems. In: Proc. of the 16th IEEE Computer Security Foundations Workshop, pp. 75–88 (2003)

[HR04]   Huth, M., Ryan, M.: Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, Cambridge (2004)

[HS04]   Hughes, D., Shmatikov, V.: Information hiding, anonymity and privacy: a modular approach. Journal of Computer Security (2004)

[HS06]   Hogben, G., Sommer, D.: A meta-data and reasoning framework for open assertion and evidence exchange and query. Technical Report RZ 3674, IBM Research, Zurich Research Laboratory (report jointly done with Joint Research Centre) (October 2006), http://domino.research.ibm.com/ library/cyberdig.nsf/index.html

[hsq07]   hsqldb (2007), http://hsqldb.org/

[IBM04]   IBM. The enterprise privacy authorization language (epal) (2004), http://www.zurich.ibm.com/security/enterprise-privacy/epal/

[IEE98]   IEEE Standards Association. 829-1998 IEEE Standard for Software Test Documentation. Technical report (1998)

[IH07]   Iachello, G., Hong, J.: End-User Privacy in Human-Computer Interaction. Foundations and Trends® in Human–Computer Interaction 1, 1–137 (2007)

[Inc06]   Microsoft Inc. Privacy Guidelines for Developing Software Products and Services, October 10, Version 2.1 (2006)

[Int99]   International Organization for Standardization. Information tech-
             nology – open systems interconnection – evaluation criteria for in-
             formation technology security. ISO/IEC 15408 (1999)

[ISO06]   ISO/IEC. Iso/iec 15408: Criteria for information technology secu-
             rity evaluation, v3.1, part 2: Security functional components. ACM
             Transactions on Computer Systems, p. 314 (2006)

[JEL03]   Johnston, J., Eloff, J.H.P., Labuschagne, L.: Security and Human
             Computer Interfaces. Computers & Security 22(8), 675 (2003)

[JJC01]   Jeanneret, J.P., Jacquet-Chiffelle, D.O.: How to protect patient's
             rigottes to medical secret in official statistics. In: Information Se-
             curity Solutions Europe Conference (ISSE), June 2001, pp. 26–28
             (2001)

[JtM00]   Jendricke, U., Gerd Tom Markotten, D.: Usability meets Security -
             The Identity-Manager as your Personal Security Assistant for the
             Internet. In: Proceedings of the 16th Annual Computer Security
             Applications Conference (December 2000),
             http://tserv.iig.uni-freiburg.de/telematik/forschung/
             projekte/kom_technik/atus/publications.html

[Jup01]   Jupiter. Trust online: Barrier today, strength tomorrow (April 2001)

[Jus06]   Juschka, A.: Analyse und Konzeptentwicklung für die Integra-
             tion von Reputationen in die identitätsunterstützte Lernumgebung
             BluES'n. Diploma thesis, Technische Universität Dresden (2006)

[Kal05]   Kalai, Y.T.: Smooth projective hashing and two-message oblivious
             transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494,
             pp. 78–95. Springer, Heidelberg (2005)

[KBM+03]   Abou El Kalam, A., Benferhat, S., Miège, A., El Baida, R., Cuppens,
             F., Saurel, C., Balbiani, P., Deswarte, Y., Trouessin, G.: Orbac:
             Organization based access control. In: IEEE International Workshop
             on Policies for Distributed Systems and Networks (POLICY), pp.
             120–131. IEEE Comp. Society Press, Los Alamitos (2003)

[KD05]   Abou El Kalam, A., Deswarte, Y.: Privacy requirements imple-
             mented with a javacard. In: 21st IEEE Annual Computer Security
             Applications Conference (ACSAC 2005), pp. 479–488. IEEE Comp.
             Society Press, Los Alamitos (2005)

[KD06]   Abou El Kalam, A., Deswarte, Y.: Sensitive data anonymization. IA
             Newsletter – The Newsletter for Information Assurance Technology
             Professionals 9(2), 8–12 (2006), http://iac.dtic.mil/iatac/

[KFF+07]   Kohlweiss, M., Faust, S., Fritsch, L., Gedrojc, B., Preneel, B.: Ef-
             ficient Oblivious Augmented Maps: Location-Based Services with a
             Payment Broker. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS,
             vol. 4776, pp. 77–94. Springer, Heidelberg (2007)

[KH04]   Köpsell, S., Hillig, U.: How to achieve blocking resistance for ex-
             isting systems enabling anonymous web surfing. In: Proceedings of
             the Workshop on Privacy in the Electronic Society (WPES 2004),
             Washington, DC, USA (October 2004)

[Kob02]   Kobsa, A.: Personalized hypermedia and international privacy. Com-
             munications of the ACM 45(5), 64–67 (2002),
             http://www.ics.uci.edu/~kobsa/papers/2002-CACM-kobsa.pdf

[Kob03]   Kobsa, A.: A component architecture for dynamically managing privacy constraints in personalized web-based systems. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 177–188. Springer, Heidelberg (2003)

[Kol99]   Kollock, P.: The production of trust in online markets. Advances in Group Processes 16 (1999)

[KSW02a]   Karjoth, G., Schunter, M., Waidner, M.: Privacy-enabled services for enterprise. In: DEXA 2002. IEEE, Los Alamitos (2002)

[KSW02b]   Karjoth, G., Schunter, M., Waidner, M.: Privacy-enabled services for enterprises. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, Springer, Heidelberg (2002)

[KSW03]   Karjoth, G., Schunter, M., Waidner, M.: Platform for enterprise privacy practices: Privacy-enabled management of customer data. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 69–84. Springer, Heidelberg (2003)

[Kum96]   Kumar, N.: The power of trust in manufacturer-retailer relationships. Harvard Business Review, pp. 92–106 (1996)

[LGF00]   Li, N., Grosof, B.N., Feigenbaum, J.: A practically implementable and tractable delegation logic. In: Proc. of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp. 27–42 (June 2000)

[Lip05]   Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 314–328. Springer, Heidelberg (2005)

[LRSW99]   Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems (Extended abstract). In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, p. 184. Springer, Heidelberg (2000)

[Luh79]   Luhmann, N.: Trust as a reduction of complexity. In: Trust and Power: Two works by Niklas Luhmann, pp. 24–31. John Wiley & Sons, New York (1979)

[Lys02]   Lysyanskaya, A.: Signature schemes and applications to cryptographic protocol design. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (September 2002)

[MB03]   Schunter, M., Backes, M., Pfitzmann, B.: A toolkit for managing enterprise privacy policies. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 162–180. Springer, Heidelberg (2003)

[MD05a]   Melchor, C.A., Deswarte, Y.: pmix: Untraceability for small hiding groups. In: NCA, pp. 29–40. IEEE Computer Society, Los Alamitos (2005)

[MD05b]   Murdoch, S.J., Danezis, G.: Low-Cost Traffic Analysis of Tor. In: Proceedings of the 2005 IEEE Symposium on Security and Privacy, Oakland, CA, USA. IEEE CS, Los Alamitos (2005)

[MD06]   Melchor, C.A., Deswarte, Y.: From dc-nets to pmixes: Multiple variants for anonymous communications. In: NCA, pp. 163–172. IEEE Computer Society, Los Alamitos (2006)

[Mer88]   Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 369–378. Springer, Heidelberg (1988)

[Mey93]   Meyer, B.: Systematic concurrent object-oriented programming. Communications of the ACM 36(9), 56–80 (1993)

[MIT04]   Moulinos, K., Iliadis, J., Tsoumas, V.: Towards secure sealing of privacy policies. Information Management & Computer Security 12(4), 350–361 (2004)

[MO04]   Mahler, T., Olsen, T.: Reputation systems and data protection law. In: Cunningham, P., Cunningham, M. (eds.) eAdoption and the Knowledge Economy. Issues, Applications, Case Studies, pp. 180–187. IOS Press, Amsterdam (2004)

[Mon04]   Casassa Mont, M.: Dealing with privacy obligations: Important aspects and technical approaches. In: Katsikas, S.K., López, J., Pernul, G. (eds.) TrustBus 2004. LNCS, vol. 3184, pp. 120–131. Springer, Heidelberg (2004)

[Mon05]   Casassa Mont, M.: A System to Handle Privacy Obligations in Enterprises. HP Labs Technical Report, HPL-2005-180 (2005)

[MOV96]   Menezes, A.J., Van Oorshot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)

[Mui03]   Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD Thesis, Massachusetts Institute of Technology (2003)

[MyS07]   MySQL: The world's most popular open source database (2007), http://www.mysql.com/

[MZ07]   Murdoch, S.J., Zieliński, P.: Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 167–183. Springer, Heidelberg (2007)

[Net04]   Reputation Research Network. Online papers on reputation and reputation research (2004), http://databases.si.umich.edu/reputations/index.html

[Nie99]   Nielsen, J.: Trust or bust: Communicating trustworthiness in web design. Jacob Nielsen's Alertbox (1999), http://www.useit.com/alertbox/990307.html

[Nie04]   Nielsen, J.: Jacob Nielsen's Alertbox, User Education Is Not the Answer to Security Problems (October 25, 2004), http://www.useit.com

[Nis99]   Nissenbaum, H.: Can trust be secured online? a theoretical perspective. Etica e Politica (2) (1999)

[NLW05]   Ni, J., Li, N., Winsborough, W.H.: Automated trust negotiation using cryptographic credentials. In: Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005), Alexandria, VA, USA (November 2005)

[NP99a]   Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: 31st ACM STOC, Atlanta, Georgia, USA, May 1-4, pp. 245–254. ACM Press, New York (1999)

[NP99b]   Naor, M., Pinkas, B.: Oblivious transfer with adaptive queries. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 573–590. Springer, Heidelberg (1999)

[NP01]   Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: 12th SODA, Washington, DC, USA, January 7-9, pp. 448–457. ACM-SIAM (2001)

[OFB97]   Orr, D.P., Dennis Fortenberry, J., Blythe, M.J.: Validity of self-reported sexual behaviors in adolescent women using biomarker outcomes. Sexually Transmitted Diseases 24(5), 261–266 (1997)

[OK04]   Ogata, W., Kurosawa, K.: Oblivious keyword search. J. Complexity 20(2-3), 356–371 (2004)

[Org80]   Organization for Economic Co-operation and Development. OECD guidelines on the protection of privacy and transborder flows of personal data (1980), http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_119820_1_1_1,00.html

[OS05]   Ostrovsky, R., Skeith III., W.E.: Private searching on streaming data. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 223–240. Springer, Heidelberg (2005)

[ØS06]   Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy. IEEE CS, Los Alamitos (2006)

[Ost01]   Osterwalder, D.: Trust through evaluation and certification? Social Science Computer Review 19(1), 32–46 (2001)

[P3P]   P3P. Implementation report, http://www.w3.org/P3P/implementation-report.html

[PA09]   Pearson, S., Allison, D.: A model-based privacy compliance checker. Special Issue on the Protection of Privacy in E-Business. IGI Global (April 2009)

[Pai99]   Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

[Pea03]   Pearson, S.: A trusted method for self-profiling in e-commerce. In: Falcone, R., Barber, S.K., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 177–193. Springer, Heidelberg (2003)

[Pea06]   Pearson, S.: Towards automated evaluation of trust constraints. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 252–266. Springer, Heidelberg (2006)

[PFHB06]   Pettersson, J.S., Fischer-Hübner, S., Bergmann, M.: Outlining Data Track: Privacy-friendly Data Maintenance for End-users. In: Proceedings of the 15th International Conference on Information Systems Development (ISD 2006), Budapest (August 31-September 2, 2006)

[PFHD+05]   Pettersson, J.S., Fischer-Hübner, S., Danielsson, N., Nilsson, J., Bergmann, M., Clauß, S., Kriegelstein, T., Krasemann, H.: Making PRIME Usable. In: Symposium on Usable Privacy and Security. Carnegie Mellon University, Pittsburgh (July 2005)

[PFHPCM06]   Pettersson, J.S., Fischer-Hübner, S., Pearson, S., Casassa-Mont, M.: How Ordinary Internet Users can Have a Chance to Influence Privacy Policies. In: Short Paper Proceedings of the 4th Nordic Conference on Human-Computer Interaction - NordiCHI 2006, October 4-18. ACM Press, Oslo (2006)

[Pfi90]   Pfitzmann, A.: *Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz*. Informatik-Fachberichte, vol. 234. Springer, Heidelberg (1990)

[PH06]   Pfitzmann, A., Hansen, M.: Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology. Draft, version 0.28 (May 2006)

[PH10]   Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. Technical report, Technische Universität Dresden, v0.34 (August 2010), `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf`

[PK03]   Patrick, A.S., Kenny, S.: From Privacy Legislation to Interface Design: Implementing Information Privacy in Human-Computer Interactions. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 107–124. Springer, Heidelberg (2003)

[PKHvB02]   Patrick, A.S., Kenny, S., Holmes, C., van Breukelen, M.: Human Computer Interaction. In: van Blarkom, Borking, O. (eds.) Handbook for Privacy and Privacy-Enhancing Technologies, PISA project, ch. 12 (2002), `http://www.andrewpatrick.ca/pisa/handbook/handbook.html`

[PP06a]   Panchenko, A., Pimenidis, L.: Towards Practical Attacker Classification for Risk Analysis in Anonymous Communication. In: Leitold, H., Markatos, E.P. (eds.) CMS 2006. LNCS, vol. 4237, pp. 240–251. Springer, Heidelberg (2006)

[PP06b]   Panchenko, A., Pimenidis, L.: Trust-Based Collusion Resistant Directory Service. In: 8th International Symposium on System and Information Security SSI 2006, Sao Paulo, Brazil, November 8-10 (2006)

[PP07]   Panchenko, A., Pimenidis, L.: Using Trust to Resist Censorship in the Presence of Collusion. In: Proceedings of the 22th IFIP TC-11 International Information Security Conference (IFIP SEC 2007), Sandton, South Africa (May 2007)

[PPR08]   Panchenko, A., Pimenidis, L., Renner, J.: Performance Analysis of Anonymous Communication Channels Provided by Tor. In: Proceedings of the Third International Conference on Availability, Reliability and Security (ARES 2008), Barcelona, Spain (March 2008)

[PRIa]   Prime project web site (2008), `http://www.prime-project.eu`

[Prib]   Privoxy: Filtering Web Proxy, `http://www.privoxy.org`

[Pro04]   PRIME Project. Requirements Version 0 – Part 3: Application Requirements (2004)

[PS08]   Pingel, F., Steinbrecher, S.: Multilateral secure cross-community reputation systems for internet communities. In: Furnell, S.M., Katsikas, S.K., Lioy, A. (eds.) TrustBus 2008. LNCS, vol. 5185, pp. 69–78. Springer, Heidelberg (2008)

[QBAB98]   Quantin, C., Bouzelat, H., Allaert, F., Benhamiche, A.: How to ensure data security of an epidemiological follow-up. International Journal of Medical Informatics 49(1), 117–122 (1998)

[R(997)]   Recommendations R(97)5. On the protection of medical data banks. Strasbourg, Council of Europe (1997)

[Rab81]   Rabin, M.O.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory (1981)

[Ras00]   Raskin, J.: The Humane Interface - New Directions for Designing Interactive Systems. ACM Press, New York (2000)

[Ray00]   Raymond, J.-F.: Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 10–29. Springer, Heidelberg (2001)

[Ren60]   Renyi, A.: On measures of entropy and information. In: Fourth Berkeley Symposium Math. Statist. and Prob., pp. 547–561 (1960)

[Res97]   Varian, H.R., Resnik, P.: Recommender systems. Communications of ACM (1997), http://www.acm.org/pubs/cacm/MAR97/resnick.html

[RHCMP05]   Roessler, T., Hogben, G., Casassa Mont, M., Pearson, S.: Position paper: Rule language requirements for privacy-enabled identity management, http://www.w3.org/2004/12/rules-ws/paper/59/

[Rhe93]   Rheingold, H.: Virtual Community. Homesteading on the Electronic Frontier. HarperTrade (1993)

[RM04]   Ribeiro, R.C., Martins, E.: A Strategy to Improve Component Testability without Source Code, vol. P-58, pp. 47–62. Kóllen Druck+Verlag GmbH, Bonn (2004)

[RP02]   Rennhard, M., Plattner, B.: Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002), Washington, DC, USA (November 2002)

[RR98]   Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security 1(1) (June 1998)

[Rub93]   Rubin, D.B.: Discussion: Statistical disclosure limitation. Journal of Official Statistics 9, 462–468 (1993)

[RZN$^+$05] Ryutov, T., Zhou, L., Neuman, C., Leithead, T., Seamons, K.E.: Adaptive trust negotiation and access control. In: Proc. of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, Sweden (June 2005)

[SA02]   Schunter, M., Ashley, P.: The platform for enterprise privacy practices. IBM Zurich Research Laboratory (2002)

[SAB$^+$]   Subrahmanian, V.S., Adali, S., Brink, A., Lu, J.J., Rajput, A., Rogers, T.J., Ross, R., Ward, C.: A heterogeneous reasoning and mediator system, http://www.cs.umd.edu/projects/hermes

[SBS02a]   Sherwood, R., Bhattacharjee, B., Srinivasan, A.: P5: A protocol for scalable anonymous communication. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy (May 2002)

[SBS02b]   Sherwood, R., Bhattacharjee, B., Srinivasan, A.: P5: A protocol for scalable anonymous communication. In: IEEE Symposium on Security and Privacy (2002)

[Sch91]   Schnorr, C.P.: Efficient signature generation for smart cards. Journal of Cryptology 4(3), 239–252 (1991)

[Sch96]   Schneier, B.: Applied Cryptography, 2nd edn. John Wiley & Sons, New York (1996)

[SD01]   Samarati, P., De Capitani di Vimercati, S.: Access control: Policies, models, and mechanisms. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, p. 137. Springer, Heidelberg (2001)

[SD02]	Serjantov, A., Danezis, G.: Towards an Information Theoretic Metric for Anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 41–53. Springer, Heidelberg (2003)
[SDOF07]	Schechter, S.E., Dhamija, R., Ozment, A., Fischer, I.: The Emperor's New Security Indicators: An evaluation of web site authentication and the effect of role playing on usability studies. In: IEEE Symposium on Security and Privacy, May 20-27, Oakland, California (2007)
[Sha48]	Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
[Sha79]	Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery 22(11), 612–613 (1979)
[Sim68]	Simmel, G.: Soziologie, 5th edn., Berlin (1968)
[SK03]	Steinbrecher, S., Köpsell, S.: Modelling Unlinkability. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 32–47. Springer, Heidelberg (2003)
[Som04]	Sommer, D.: PRIME Architecture V0. Technical report, PRIME Project (2004), https://www.prime-project.eu/prime_products/reports/arch/
[Som05]	Sommer, D.: PRIME Architecture V1. Technical report, PRIME Project (2005), https://www.prime-project.eu/prime_products/reports/arch/
[Som08]	Sommer, D.: PRIME Architecture V3. Technical report, PRIME Project (2008), https://www.prime-project.eu/prime_products/reports/arch/
[SPC05]	Pearson, S., Casassa Mont, M., Crane, S.: Persistent and dynamic trust: Analysis and the related impact of trusted platforms. In: Issarny, V., Herrmann, P., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 355–363. Springer, Heidelberg (2005)
[Spi03]	Spiekermann, S.: Die Konsumenten der Anonymität – Wer nutzt Anonymisierungsdienste? Datenschutz und Datensicherheit 27(3) (2003)
[SS96]	Schneider, S., Sidiropoulos, A.: CSP and anonymity. In: Martella, G., Kurth, H., Montolivo, E., Hwang, J. (eds.) ESORICS 1996. LNCS, vol. 1146, pp. 198–218. Springer, Heidelberg (1996)
[SS99]	Syverson, P.F., Stubblebine, S.G.: Group principals and the formalization of anonymity. In: World Congress on Formal Methods (1), pp. 814–833 (1999)
[SS03]	Serjantov, A., Sewell, P.: Passive Attack Analysis for Connection-Based Anonymity Systems. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 116–131. Springer, Heidelberg (2003)
[SS05]	Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artif. Intell. Rev. 24(1), 33–60 (2005)
[Ste98]	Stern, J.P.: A new and efficient all-or-nothing disclosure of secrets protocol. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 357–371. Springer, Heidelberg (1998)

[Ste06]   Steinbrecher, S.: Design options for privacy-respecting reputation systems within centralised internet communities. In: Proceedings of IFIP Sec 2006, 21st IFIP International Information Security Conference: Security and Privacy in Dynamic Environments (May 2006)

[Swe02]   Sweeney, L.: k-Anonymity: A model for protecting privacy. International Journal on Uncertainity, Fuzziness and Knowledge-based Systems 10(5), 557–570 (2002)

[SWW97]   Seamons, K.E., Winsborough, W., Winslett, M.: Internet credential acceptance policies. In: Proc. of the Workshop on Logic Programming for Internet Applications, Leuven, Belgium (July 1997)

[SWY01]   Seamons, K., Winslett, M., Yu, T.: Limiting the disclosure of access control policies during automated trust negotiation. In: Proc. of the Network and Distributed System Security Symposium (NDSS 2001), San Diego, CA, USA (April 2001)

[Syv03]   Syverson, P.: The paradoxical value of privacy. In: 2nd Annual Workshop on Economics and Information Security (WEIS 2003), College Park MD (2003)

[TB06]   Tabriz, P., Borisov, N.: Breaking the Collusion Detection Mechanism of MorphMix. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 368–383. Springer, Heidelberg (2006)

[Tec]   Technische Universität Dresden. Dresden identity management (drim)

[THV04]   Tóth, G., Hornák, Z., Vajda, F.: Measuring anonymity revisited. In: Liimatainen, S., Virtanen, T. (eds.) Proceedings of the Ninth Nordic Workshop on Secure IT Systems, Espoo, Finland, pp. 85–90 (November 2004)

[Tru03]   Trusted Computing Group. Tcg main specification (2003), http://www.trustedcomputinggroup.org

[Tru04]   Trusted Computing Group. Tcg specification architecture overview (2004), http://www.trustedcomputinggroup.org

[Tru06]   Trusted Computing Group. Tpm specification (2006), http://www.trustedcomputinggroup.org

[Tur02]   Turner, C.W.: The online experience and consumers' perceptions of e-commerce security. In: Proc. Human Factors and Ergonomics Society 46th Annual Meeting (2002)

[Tur03]   Turner, C.W.: How do consumers form their judgments of the security of e-commerce web sites. In: Proc. Workshop on HCI and Security Systems, CHI 2003 (2003)

[TZY01]   Turner, C.W., Zavod, M., Yurcik, W.: Factors that affect the perception of security and privacy of e-commerce websites. In: Proc. 4th International Conference on Electronic Commerce Research (2001)

[UDH98]   Universal Declaration of Human Rights (1998), http://www.un.org/Overview/rights.html (visited December 2007)

[UDH03]   Documentation of Internet Filtering Worldwide (2003), http://cyber.law.harvard.edu/filtering/ (visited March 2006)

[vABH03]   von Ahn, L., Bortz, A., Hopper, N.J.: k-anonymous message transmission. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM Conference on Computer and Communications Security, pp. 122–130. ACM, New York (2003)

[Vas05]   Vasic, O.: Using tricryption to protect privacy. Technical report, Tech. Report ERUCES Data Security (2005)

[ver]   Verisign identity protection, http://www.verisign.com/products-services/security-services/identityprotection/index.html

[Ver01]   Verheul, E.: Self-blindable credential certificates from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 533–551. Springer, Heidelberg (2001)

[Vie95]   Vienneau, R.L.: The Present Value of Software Maintenance. Journal of Parametrics 25(1), 18–36 (1995)

[Vos04]   Voss, M.: Privacy preserving online reputation systems. In: International Information Security Workshops, pp. 245–260. Kluwer, Dordrecht (2004)

[VWW02]   Vetterling, M., Wimmel, G., Wisspeintner, A.: Secure systems development based on the common criteria: the PalME project. SIGSOFT Softw. Eng. Notes 27(6), 129–138 (2002)

[W3C02]   World Wide Web Consortium. A P3P Preference Exchange Language 1.0 (APPEL1.0) (April 2002), http://www.w3.org/TR/P3P-preferences/

[W3C03]   W3C. Extensible markup language (xml) (2003), http://www.w3.org/XML/

[W3C04]   W3C. Resource description framework (rdf) (2004), http://www.w3.org/RDF/

[Wai90]   Waidner, M.: Unconditional sender and recipient untraceability in spite of active attacks. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 302–319. Springer, Heidelberg (1990)

[WCJS97]   Winslett, M., Ching, N., Jones, V., Slepchin, I.: Assuring security and privacy for digital library transactions on the web: Client and server security policies. In: Proc. of the 4th International Forum on Research and Technology Advances in Digital Libraries (ADL 1997), Washington, DC, USA (May 1997)

[Wen96]   Wenger, E.: Communities of Practice: The Social Fabric of a Learning Organization. Health Forum Journal 43(4), 20–25 (1996)

[WML06]   Wu, M., Miller, R.C., Little, G.: Web Wallet: Preventing phishing attacks by revealing user intentions. In: Symposium On Usable Privacy and Security (SOUPS) 2006, July 12-14 (2006); Available in ACM Digital Library

[Wor02]   World Wide Web Consortium. Platform for privacy preferences (P3P) project (April 2002), http://www.w3.org/P3P/

[WP90]   Waidner, M., Pfitzmann, B.: The dining cryptographers in the disco: Unconditional sender and recipient untraceability with computationally secure serviceability. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 690–690. Springer, Heidelberg (1990)

[WP608]   WP6.1. Hci guidelines. In: Pettersson, J.S. (ed.) HCI Guidelines, PRIME Deliverables number D6.1.f (February 2008), https://www.prime-project.eu/prime_products/reports/arch/

[WSJ00]   Winsborough, W., Seamons, K.E., Jones, V.: Automated trust negotiation. In: Proc. of the DARPA Information Survivability Conference & Exposition (DISCEX 2000), Hilton Head Island, SC, USA (January 2000)

[WT99]   Whitten, A., Tygar, J.D.: Why Jonny Can't Encrypt: A Usability Evaluation of PGP 5.0. In: Proceedings of the Ninth USENIX Security Symposium (1999)

[YC04]   Yan, Z., Cofta, P.: A mechanism for trust sustainability among trusted computing platforms. In: López, J., Katsikas, S.K., Pernul, G. (eds.) TrustBus 2004. LNCS, vol. 3184, pp. 11–19. Springer, Heidelberg (2004)

[YF04]   Yao, G., Feng, D.: A New k-Anonymous Message Transmission Protocol. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 388–399. Springer, Heidelberg (2005)

[Yun03]   Yung, M.: Trusted computing platforms: The good, the bad, and the ugly. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 250–254. Springer, Heidelberg (2003)

[YW03]   Yu, T., Winslett, M.: A unified scheme for resource protection in automated trust negotiation. In: Proc. of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA (May 2003)

[YWS01]   Yu, T., Winslett, M., Seamons, K.E.: Interoperable strategies in automated trust negotiation. In: Proc. of the 8th ACM Conference on Computer and Communications Security (CCS 2001), Philadelphia, PA, USA (November 2001)

[YWS03]   Yu, T., Winslett, M., Seamons, K.E.: Supporting structured credentials and sensitive policies trough interoperable strategies for automated trust. ACM Transactions on Information and System Security (TISSEC) 6(1), 1–42 (2003)

Part IV

PRIME Applied

23

Introduction

Pete Bramhall

Hewlett-Packard Laboratories

A key component of PRIME's holistic approach to its objective of showing that user-controlled identity management is possible was the work to develop and evaluate application-level prototypes that employ PRIME principles and technologies. The objectives behind this work were twofold:

- To validate, in specific real-life environments, the approach, architecture and technology of PRIME.
- To provide learnings from this process that could be fed into other project activities, in order to enable improvements.

The work was undertaken as the final stage of two of the PRIME project's development cycles, building on previous work and providing input into the definitions of subsequent work. Its output was:

- A set of three initial application prototypes and their evaluation.
- A set of two final application prototypes and their evaluation

The final application prototypes were derived from two of the initial application prototypes.

In each case, the method employed was to

- Design and implement prototypes for the application scenarios,
- Perform limited trials of these, and
- Analyse and evaluate the results from Human-Computer Interaction (HCI), assurance, legal, economic and social/cultural aspects, and report these.

Careful thought was given to the selection of the application scenarios, in order that these would provide a rich and sufficiently wide set of challenges across the range of requirements identified by the project. On this basis, the initial applications were chosen to be

Collaborative eLearning (CeL). This scenario offers
- A very rich set of roles and actors
- A diverse set of types of interaction

Location Based Services (LBS). This scenario offers
- A complex services side, with multiple parties and systems
- Complex data flows

Airport Security Controls (ASC). This scenario offers
- The opportunity to test the compatibility and integration of PRIME principles with IdM processes and user/service interactions

In each case, the existence of the application prototype provided its evaluators with the opportunity to discover and investigate aspects of the use of PETs which would apply to a wider set of technical approaches than just that of PRIME. This was particularly true of the social/cultural, HCI and economic aspects.

The mapping of the set of requirements against the chosen application scenarios is shown in Table 23.1.

Table 23.1 Mapping of requirements to prototypes

CeL	LBS	ASC	Requirement
CeL	LBS	ASC	Data minimisation
CeL	LBS	ASC	Partial identities
CeL	LBS	ASC	Multiple pseudonyms, multiple types
CeL	LBS	ASC	Support for anonymous behaviour
CeL			User control over partial identity profile
CeL			Enable building of reputation
CeL	LBS		Rights management / access control
CeL			Role/rights switching
CeL	LBS		Context awareness and management
CeL	LBS		Privacy policies
CeL	LBS	ASC	Credentials provisioning and management
	LBS		Support for dispute resolution
CeL	LBS	ASC	Data integrity
	LBS	ASC	Support for IdM obligations

It was important that the trials of the application prototypes be limited in scope to those that could be implemented without requiring investment in, or disruption to, the supporting communication and storage network infrastructure. It was also important that the trials support organisation be very limited, e.g., without a dedicated helpdesk, such that the necessary support could be provided by the application prototype development teams themselves without major impact to their other work. These limitations were self-imposed by PRIME for cost-containment reasons.

It was also important that the engineering of the application prototypes be limited to that which was strictly necessary to achieve their objectives. The project wished to avoid directing its resources to activities that were aimed at providing commercial-grade qualities instead of addressing research questions. Consequently, the prototypes were not "polished", and the resulting roughness provided rich material for the evaluators, especially in the HCI and social/cultural aspects.

Taking the above into account, the application prototypes should be viewed more as demonstrators rather than as early versions of deployable systems.

Although the development of the PRIME Integrated Prototype included the creation of some application-level demonstrator code that provided a context for testing and evaluating the Integrated Prototype, the Collaborative eLearning and Location Based Services application prototypes provided the first real opportunity for validation of the PRIME technical approach, as their IdM subsystems used the Integrated Prototype as a toolbox for the necessary functions.

The initial versions of the three application prototypes were designed and built in 2005, using the first versions of the PRIME Architecture and Integrated Prototype as their technical basis. They were evaluated at the end of 2005 and the evaluation report was delivered in February 2006. Its main points were:

- These initial application prototypes did provide a sufficient basis for the first detailed evaluations to be made of the PRIME technical approach from multiple viewpoints.
- In the final versions of the application prototypes, the following are recommended:
    - Simplify the choices for the user, but also provide for greater individual control over PII use, including the possibility of using real names instead of pseudonyms.
    - Provide a privacy-respecting logging capability.
    - Pay more attention to good information security design principles in general.

Project budget limitations required that only two final application prototypes be developed and evaluated, and it was decided that these should be for Collaborative eLearning and Location Based Services, as the IdM subsystems of these make the most use of PRIME technology. Accordingly, these final application prototypes employed the second versions of the PRIME Integrated Prototype.

The principal enhancements within the final version of the Collaborative eLearning application prototype, as compared to its initial version, were:

- Improvements and extensions in partial identities, pseudonyms and aliases,
- Enhanced access control capabilities,
- Enhanced context management,

- Improvements in the user interface, and
- More communication functions.

The principal difference between the initial and final versions of the Location Based Services application is that the former provided a pull-based service (a pharmacy finder) and the latter provided a push-based service (a service that warns of high pollen levels, as both these and the user's location change). These different service approaches require different co-operating entities, data flows and user inputs.

The second cycle of application prototypes were developed and evaluated in 2007. The Collaborative eLearning prototype was completed in July 2007, and its evaluation was finished in November 2007. The Location Based Services prototype was developed in two stages. The first stage yielded a pre-release version, in which much of the functionality was present as a mock-up, but which was suitable for an initial evaluation of the non-technical aspects and for planning the full evaluation of the full version. The second stage yielded the full version in September 2007, and its evaluation was finished in December 2007.

The complete evaluation of both these final application prototypes was delivered in February 2008. Its principal conclusions are that:

- The APs have provided a clear demonstration of the technical feasibility of the PRIME approach to providing privacy, and that they form assets which are valuable to privacy specialists working in the various disciplines which are relevant to its further development and exploitation.
- No fundamental issues have been found regarding compliance to European law.
- There remain significant operational issues relating to the adoption of PETs, and the PRIME approach in particular, by organizations and individuals.
- Great attention must be paid by system designers to provide users with all the information they need about why personal information is required and what will happen to it, which entities will see how much of it, who they are and how they are related.
- When user-centric PETs are employed, great attention must be paid by system designers to educating users about concepts such as anonymisation and partial identities and initiating them into the use of these, for example by means of a very clear and rich Help function and instructive tool tips.

24

Collaborative E-Learning

Katja Liesebach, Elke Franz, Anne-Katrin Stange, Andreas Juschka,
Katrin Borcea-Pfitzmann, Alexander Böttcher, and Hagen Wahrig

Technische Universität Dresden

In the following chapter a short overview about the collaborative eLearning application prototype BluES'n is given. Starting by emphasising its need and potentials for PRIME, the integrated and realised privacy-enhancing components and functionalities are described. A summarising section points out lessons learnt when integrating PRIME into the application.

24.1 The Collaborative eLearning System BluES'n

24.1.1 Democratisation of an eLearning Environment

"Everyone is allowed to do everything – in the frames of generally-agreed rules and directives" – this statement depicts the idea for designing an eLearning environment where every user gets access to all functionalities provided by the environment (cf. [BPL07], [BPLW05]). Each user should have the possibility to read and annotate learning contents, to generate own contents as well as to structure them according to his preferences. Furthermore, the user should also be supported to perform those actions together with other users of the eLearning environment. This implies the availability of possibilities for dynamic group building as well as for the non-restricted use of collaboration and communication modules. Consequently, it is imperative to refrain from the traditional and rigid approach of role handling in eLearning. A system is to be designed that poses the individual user and user groups as well as

their interests and competencies in the centre of the working and learning environment. Thereby, all functions have to be provided to efficiently achieve learning and working objectives.

Comparable to traditional real-world educational scenarios where various working processes take place in different rooms and areas, users are provided an eLearning environment with according working areas – so-called workspaces. Such workspaces are "equipped" with all necessary functionality and means for an objective-oriented coping with tasks and interactions.

The metaphor of a "workspace" for an objective-oriented partitioning of the eLearning environment is used to facilitate different fields of activity. Two types of workspaces can be identified: In the centre of *Shared Workspaces* are the corporate work and communication of the participants. In contrast, *Personal Workspaces* represent users' individual working environments, i.e., every user has her own individual workspace – the Personal Workspace – which represents, therefore, a special form of a shared workspace.

Following the democracy approach, each user has the possibility to create and configure new shared workspaces. A newly created workspace is equipped in accordance with the requirements of the task that should be elaborated as well as with the individual characteristics of the working community. These settings are performed by the corresponding workspace owner. The main characteristics of an individual workspace are *functional modules*, the *contents* to be worked on and *users and their roles*.

The functionality necessary for learning and working in a workspace is provided by so-called *functional modules*. That way, the users get access to very different educational functions, such as tools that allow for collaborative elaboration of knowledge and documents, for structuring contents and for communication as well as for interactions between users.

Another element characterising a specific workspace is the *content*. It is the working basis since it objectifies the knowledge that is created and enhanced during the accomplishment of work in the workspaces.

Finally, the efficient proceeding of learning and working processes requires roles describing intentions and responsibilities. According to real-world scenarios, roles become an important factor of social life when people are interacting, i.e., no global pre-assignment of roles is intended. Instead, two different kinds of roles are considered: Administrative and functional roles. By means of *administrative roles*, general possibilities of users and their rights to access resources in the corresponding workspace are described. Beside the *workspace owner*, which has extensive rights in the workspace, we consider *participants* and *guests* as further administrative role instances. While *participants* are allowed to act and take active part in a specific workspace, highly restricted access to offered contents and functionalities are provided to *guests*. In contrast to administrative roles, the instances of functional roles, e.g., tutor, author, moderator, describe privileges and obligations of users. They are used to communicate other users of this workspace their own status, role-related

assignments, and responsibilities while working with a corresponding functional module.

Summarising, the support of collaborative and cooperative learning scenarios is in the focus of the described concept. Depending on the current task and situation, learners become enabled to jointly elaborate knowledge. One of the major aims of such collaborative eLearning environments is to foster users' self-determination with respect to learning methods and styles by allowing for the creation of new working areas and groups as well as for the possibility to re-organize learning groups by the users.

24.1.2   Need for Privacy and How PRIME Helps

In traditional eLearning environments where users work under one login only, all their actions within that application can be linked. This, however, offers the possibility to create detailed user profiles: First, it is recognisable which classes and groups a learner attends. Second, all actions within a class or group can be assigned to the particular user. For example, frequency of learning sessions, average duration of processing learning modules, or results of tests can be observed. Third, this collected information allows drawing conclusions about the learner, e.g., about interests, learning speed, habits, or equipment. Users may lose reputation due to failures. This may result in a biased environment: A user may be prejudiced against other users due to bad results in other classes or due to former discussions or questions. If users are aware of these threats, they might feel to be observed and be afraid of failing. Hence, they may feel restricted and become afraid to disgrace themselves. They possibly become discouraged to ask and practice.

Despite these possible privacy risks, however, users cannot act completely anonymously within a collaborative eLearning system, i.e., performing all actions anonymously and unlinkably. Collecting and evaluating personal data such as information about users' preferences and goals is necessary, e.g., to provide assistance for users, to realise assessment, or to support collaboration between users. However, privacy issues are not sufficiently considered in current eLearning environments and especially within collaborative eLearning. As users just begin to become sensitised for privacy problems in other application areas such as eShopping, awareness of privacy threats is not yet widely established within the field of eLearning.

Results of surveys regarding privacy protection in the Internet ([INR97, WHA98, CRA99, TK04],) as well as of eLearning services [KBG04] have shown that the majority of users are concerned about the usage of their personal information while being online. Survey results of a study conducted within the project PRIME had shown that users of eLearning systems set a high value on informational self-determination [BPS07]. Furthermore, an analysis was conducted regarding data protection in Learning Management Systems (LMS) in general [Sta07]. Six state-of-the-art LMS, e.g., WebCT and Moodle, were investigated. The results show that currently used LMS support aspects of

data protection only insufficiently. The processing of personal and identifiable information is not or only in part designed transparent for users of these systems. In addition, users are not able to determine what exactly happens with their personal data. Summarising, the wish and need for informational self-determination expressed by eLearning users of the study conducted in 2006 is not yet supported in the desired degree.

Considering the acceptance of an application in the long run, handling privacy risks is a vital task [BFD+05, BDF+05b]. A known approach to preserve privacy despite the need for collecting and processing personal data is to partition this data by means of Privacy-enhancing Identity Management (PIM) [CPHH02]. Users are enabled to decide on their own which data is delivered to whom after considering the current situation. The established sub-sets of personal data are called *partial identities (pIDs)*. Since different pIDs should not be linkable except by their owner, *pseudonyms* are used as identifiers replacing the real name of the user [PH06].

The use of pIDs would enable users to be recognisable only if necessary, e.g., in order to enable reasonable discussions with others or to enable the tutor to assist them. If learners start learning in a new class, they get the possibility to work in an unbiased environment independently of results of former classes. Additionally, learners can act under different pIDs and possibly even anonymously within one and the same class. Separating activities encourages learners to feel unrestricted and, thus, to learn without pressure. Besides this separation, the explicit linking of information is needed. Users must be able to build up their own reputation by disclosing certain information. Finally, a fine-grained partitioning of information in order to enable reasonable assistance of users or evaluation while enforcing their privacy requirements is needed.

A corresponding proof-of-concept implementation is the privacy-enhanced collaborative eLearning application BluES'n[1] which was developed in the project PRIME. BluES'n follows on the one hand the approach for collaborative working and learning as described in Sec. 24.1.1 and is otherwise enhanced by mechanisms to support user's privacy while acting in the environment.

The BluES'n system gives users the opportunity to get detailed information on the processing of their personal data and determine the data processing concerning their wishes and needs. Thereby, BluES'n utilises PIM functionalities provided by the PRIME integrated prototype. It allows for working under different pIDs for controlling the dissemination of personal information. The PIM functionalities support users to manage their pIDs, comprising tasks like creating and managing pseudonyms and managing preferences about disclosure of personal data. Policies at user side can control the disclosure of personal data. Furthermore, possibilities are needed to explicitly restrict users. For example, they should be able to take examinations only once. This problem can be solved by means of *anonymous credentials* [CL01]. Anonymous credentials

---

[1] BluES'n stands for BluES like universal eEducation System privacy-enhanced.

ensure that users can demonstrate possession of certain assertions without the necessity to link this show to their user identity.

Despite these privacy-enhancing extensions, the eLearning application must still be easy and intuitive to use. If users are just overwhelmed with many new tasks, they will most probably not utilize the functionality that enables privacy-aware learning.

24.2 Intra-Application Partitioning of Personal Data

24.2.1 Necessity and General Goals

Usually, PIM is used to keep data in different applications separate from each other (inter-application partitioning). However, this approach is not sufficient for applications providing a lot of complex and/or collaborative scenarios such as BluES'n: If all actions of one user were linkable, creating detailed user profiles would become possible. Consequently, a fine-grained partitioning of personal data within the application itself, i.e., *intra-application partitioning (IAP)* is necessary [BDF+05a, FE06, FBP06]. IAP enables users to work under different pIDs within one application (Fig. 24.1).
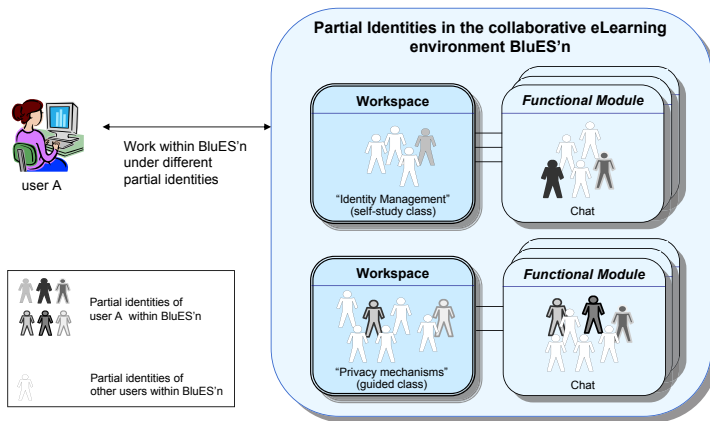
Fig. 24.1 The concept of intra-application partitioning applied to BluES'n

For usability and user acceptance reasons, we aim at supporting users in partitioning their data within the application. Thereby, we focus on two aspects:

1. Continuous *privacy awareness* (Sec. 24.4) should enable users to assess their current privacy state and, therefore, motivate them to apply IAP.

Furthermore, users will be supported in a reasonable use of IAP due to the feedback provided.

2. The system should support users in partitioning their personal data according to their preferences while they are working on other tasks. If possible, decisions should be automated. Simplifying the ongoing partitioning as much as possible should motivate users to apply IAP.

24.2.2   Concept for the Support of IAP

A context-aware component of the system (the *Decision Suggesting Module, DSM*) has the task to realise the necessary user support. Particularly, the DSM evaluates the current context and generates a suggestion regarding the decision under which pID an initiated action should be performed. All information related to an action, i.e., data requested by the server due to access control policies, data explicitly sent by the user, and information about the action itself, can be assigned to this pID. Within BluES'n, the following attributes of a user are considered:

- A pseudonym as unique identifier of a pID within the whole environment,
- A local alias as usable representation of the pseudonym [BPFP05],
- Roles describing rights and privileges of users within BluES'n, and
- Additional information helpful for supporting collaboration between users, e.g., name, address, age, and interests.

Observing actions allows to determine, e.g., when and how often a user works (under a specific pID) in a special workspace. The partitioning shall prevent others from getting a global view on all attributes assigned to a user as well as on all actions performed by the user.

The DSM evaluates a number of context features to assess the (un)linkability of an action from the point of view of other users (Tab. 24.1). The user defines rules targeted for various scenarios, e.g., for working in an authoring workspace or for participating in a chat in that workspace in order to decide whether privacy or recognition shall be supported. It depends on the aspects that should be considered by the suggestion which context features are actually evaluated. Generally, several aspects can be considered for generating the suggestion, e.g.:

- Granularity of partitioning,
- Support recognition by other users,
- Functional roles, and
- Linkability from point of view of other users.

Generally, the DSM assigns a rating to all pIDs already used within BluES'n as well as to the possibility to generate a new pID for the initiated action. If exactly one pID gets the highest rating, the user can also decide on applying this suggestion automatically. A common example is that the user performs a sequence of actions within one and the same workspace. Obviously,

Table 24.1 Context features considered by the DSM

Class of context features	Examples	Higher-level context features
Application-internal context features provided by a context monitoring component executed within BluES'n	current workspace and functional module	user's current objective, reasonable privacy preferences
	initiated action	
	pIDs currently visible for other users	potential visibility of an action for other users
Contact-related context features derived from server requests and from awareness information	pIDs of other users who can potentially observe this action	
	data required due to access control policies	increased degree of knowledge of other users about own pIDs due to necessary delivery of data
	details about data request, e.g., purpose and storage	
History-related context features derived from a history of former actions performed by the user	pIDs already used within the current workspace	increased degree of knowledge of other users about own pIDs, possibility to be recognised
	pID used for the last action performed in the current workspace	
	pIDs used for communicating with pIDs of other users currently present in the workspace	

it is reasonable in this case to use one pID for all actions instead of disturbing the user every time.

Finally, the user must select one of the pIDs for the initiated action. He should use the pID with the highest rating, but he always has the possibility to select another pID or to generate a new pID, respectively.

24.2.3 Realisation within the CeL Prototype

The DSM is called within PRIME when a decision regarding the delivery of personal data is required. It gets as input all pIDs used so far within BluES'n. The *Context Monitoring* delivers the application-internal context features; the DSM derives history-related context features from the PRIME history data base. Afterwards, it selects the appropriate configuration considering the state of the corresponding scope, e.g., a user works for the first time in this workspace/functional module.

The current prototype version only supports a granularity of partitioning considering the levels "BluES'n", "Workspace", and "Functional Module". The level influences the scope of pIDs, e.g., the level "Workspace" implies

that pIDs are mainly used within *one* workspace and, hence, personal data and actions are partitioned between different workspaces.

According to the context and based on the configuration, the DSM generates a rating for all pIDs which is a numeric value ranging from 10 (highest rating, should be preferred) to 0 (should not be used). Automatically applying the suggestion is possible as described above. Otherwise, the adapted "Send Personal Data"-Dialogue displays the suggestion (Sec. 24.4.4).

Since foreseeing all possible situations is not feasible, users must be able to intervene the application of predefined rules, especially if the suggestion should be applied automatically. Thus, users can switch off the DSM for the next action or until it is re-activated. The scope of deactivation is the respective workspace. If the DSM is switched off, any server request implies a user interaction by calling the adapted "Send Personal Data"-Dialogue.

Furthermore, in certain situations the pre-selection of a pID is of interest, e.g., during synchronous communication such as a chat session. Therefore, accordant functionalities are provided to the BluES'n user.

24.2.4   Discussion

The integration of a context-aware component allows for an easier partitioning of personal data. There are mainly three advantages: First, users can define rules regarding the selection of pIDs before they start their actual work, i.e., without pressure. Second, the evaluation of these rules during everyday work in the application ensures that suggestions are made according to the actual privacy preferences. Finally, the suggestions speed up the decision; the pIDs are grouped according to their suitability for the current situation, and the most plausible one is already highlighted and can be selected quickly. Additionally, the possibility to automate decisions increases the performance and reduces user interactions.

Future versions will consider further aspects, e.g., supporting recognition. It is a challenging task to define reasonable rules for conflicting goals like recognition and linkability. Defining rules dependent on the current role seems to be advantageous since it will be more intuitive for users.

Currently, the DSM supports IAP especially for BluES'n. However, a reasonable support for users will be needed also for other complex or collaborative applications which require IAP. Furthermore, the evaluation of contextual information can also be applied in order to support inter-application partitioning. Thus, it is a topic of future work to generalise the concepts applied to be used for other scenarios.

24.3   Policy- and Credential-Based Access Control

24.3.1   Necessity for Privacy-Enhancing Access Control

Access control mechanisms are required to proctect resources from unauthorised access. This comprises services for *identification* and *authentication*, i.e., who is allowed to access resources, i.e., data and services, and *authorisation*, i.e., to decide who is allowed to operate on these resources.

Within collaborative eLearning, access control is needed to constrain the usage of services and functionalities, to regulate access to provided contents, and to protect user data stored on the server side. Storing some user data on the server side is needed to make them available even if the corresponding user is not online in order to provide important services such as delivering awareness information to other users to support cooperative working or to enable tutors to assist learners.

Usually, in collaborative eLearning environments, access rights are assigned to certain users of a system based on so-called access control lists. However, since users in BluES'n have the possibility to work under different, primarily unlinkable pIDs addressed by unique pseudonyms, relying on traditional login/password mechanisms is not possible. In order to allow for using assigned rights under different pIDs, realising a capability-based access control is reasonable. In such a model, access to resources is granted by holding a corresponding capability, i.e., an unforgeable reference, to that object. Finally, privacy-enhancing access control requires the consideration of following issues:

- Possibility for access control even if users act under several pIDs;
- Possibility to access resources independently of the user's current pID;
- Unlinkability of pIDs must not be threatened by access control.

In order to achieve flexibility, a user should be able to use capabilities independently of pIDs. However, showing such capabilities must not threaten the unlinkability of different pIDs. That means, providing evidence to own capabilities must not be linkable to user's real identity and, furthermore, showing one capability repeatedly or in different contexts must not allow for linking different pIDs of one user. A possible way to ensure that, is to use anonymous credentials as provided by PRIME in order to express the capabilities assigned to pIDs (see [FWBBP06]).

24.3.2   Realisation within the CeL Prototype

Due to the possibility for IAP in BluES'n, the need for a fine-granular access control based on credentials as well as pseudonyms can be derived. Thus, pseudonyms, credentials, and policy mechanisms provided by PRIME are used to realise a privacy-enhancing access control within the application. In BluES'n access control takes place on the level of resources, i.e., for each

resource, such as workspaces, structure elements, and learning materials, access rights are defined. Corresponding access types are *create*, *read*, *write*, and *delete* operations, whereby an access type is the smallest entity on which access control decisions can be performed. The assignment of access rights to users, i.e., to their pIDs, is made based on the BluES'n concept of administrative roles as already described in Sec. 24.1.1. A user who creates a BluES'n resource, automatically becomes its *owner* and obtains all possible access types on that resource. An owner is able to admit other users to reuse owned resources by granting according rights to them by assigning users the according role *participant* or *guest*.

Defining access rights for workspaces is closely related to the question how administrative roles are assigned to users. Therefore, we realised the concept of *workspace access modes* in BluES'n. These modes are in general pseudonym based or credential based, i.e., access to a workspace depends either on the usage of a specified pseudonym or on showing a requested credential. The latter case implies also credentials evidencing properties and pre-requisites, such as the successful conclusion of another workspace or holding a specific certificate to access a workspace. If a workspace is created, the BluES'n server issues a credential to its owner permitting *create*, *read*, and *write* operations within this workspace. Corresponding PRIME access control policies are created on the server side specifying which evidence, i.e., which workspace credential, must be delivered by the BluES'n clients in order to get access to and execute an operation on the required resource. The owner is allowed to grant other users access to his workspace by issuing according credentials to them. Whenever a BluES'n resource is to be created within a workspace, the BluES'n server first checks whether the user owns a credential containing the workspace identifier and the necessary administrative role. If the user is allowed to create a new resource within this workspace, the resource is added and corresponding PRIME policies are created for that new resource. Instead of issuing new credentials for each new resource, the BluES'n server derives the new policy for the new resource from the policy of the workspace to which the new resource is to be added.

24.3.3   Discussion

The described access control approach based on pseudonym, policy, and credential mechanisms provided by PRIME focuses on being privacy enhancing. Users are allowed to partition their personal data, i.e., work with the application using different pIDs while their actions are unlinkable.

During the realisation of access control for BluES'n, two different models for newly-created resources of workspaces are considered:

Model I: Issuing (new) credentials and creating (new) policies.
- *Content of credential*: Reference to that resource.
- *Content of policy*: Request for credential with reference to the resource, where the resource was created.

Model II: Reusing workspace credentials and creating (new) policies.
- *Credential*: Reuse of corresponding workspace credential.
- *Content of policy*: Request for the *owner*, *guest* or *participant* credential of the workspace where the resource was created.

While Model I was already integrated and tested in the first prototype version, tests and evaluations have shown that users are overstrained by this model. The process of selecting and showing an appropriate credential in order to use the created resources has to be applied too often by the user. Consequently, users felt overstrained and were distracted from their particular work. In order to avoid disturbing the user, finally, Model II was applied. Here, users create resources in the context of an opened workspace, which will become part of the policy of the created resource. That means, the creation of new resources only leads to the definition of new policies without the need for issuing a new credential. The policy is generated on basis of the active workspace and the current administrative role the user owned during the creation process. However, the chosen approach raises still open question with respect to, e.g., the deprivation and delegation of access rights. Challenges such as guaranteeing and ensuring copyrights, usage and exploitations rights within a privacy-enhancing collaborative application are further research questions the developer team is currently working on.

24.4 Privacy-Aware and Usable Application Design

Since collaboration and communication between users are of special interest within BluES'n, providing users with awareness information is of great importance in many ways. Awareness information covers all information which is necessary to allow a user assess his current situation within the application, e.g., information about the own learning progress in comparison to other users, information about other users, information about the workspace he currently acts in. Obviously, an evaluation of user's personal data is needed to provide the necessary information. Therefore, knowing what the application is doing with his personal data gives the user transparency about the information visible to and processed by the system, which might raise his trust in and thus, acceptance of the system as well. Receiving information regarding availability and current activities of other users improves the perception of inherent collaborative opportunities. However, this kind of awareness information is also an intrusion to privacy, since building up detailed profiles of other communication partners might be possible. Due to this and PRIME's maxim "design must start from maximum privacy" special attention has to be paid to the design of a privacy-aware application. Besides the need for providing group and privacy awareness information, usability is another important requirement which has to be considered when designing for end users. Achieving given objectives in an effective, efficient and satisfying way is just as essential as a user-friendly design and handling of according user interfaces.

BluES'n addresses the mentioned requirements by appropriate interfaces and functionalities in order to provide privacy-aware but also intuitive user-interfaces. In the following sections, these approaches are presented.

24.4.1   Management of Aliases

Motivation and Description of the Idea

The usage of randomly generated and uncorrelated pseudonyms as identifiers of pIDs ensures that pseudonyms do not leak information related to the user which might allow others to link different pIDs or even to draw conclusions about the real identity of the pID's holder [BPFP05]. However, the handling and management of such pseudonyms presented by randomly-looking character strings are neither user-friendly nor usable. It is not possible for users to remember these pseudonyms easily or to recognise already known pseudonyms of other users in highly dynamic situations like a chat. Hence, a user may wish to assign additional shorthand semantics – mnemonics – to pseudonyms. A mnemonic is the presentation of a pseudonym to a user and can be of arbitrary type, e.g., an alias as textual representation, images, or sound. For BluES'n, an alias/pseudonym mapping was chosen. In order to retain the privacy and security properties, these aliases should be assigned and used locally. However, a reasonable support of local aliases is necessary in order to achieve user acceptance as well. Managing local aliases basically requires the three components

- Alias Assignment,
- Alias Improvement, and
- Alias Replacement.

First, the *Alias Assignment* component assists the user in assigning aliases to pseudonyms. Generally, aliases are assigned to own pseudonyms as well as to pseudonyms of other users. Possible aliases are contained in a local alias dictionary (LAD); assigned aliases are stored as attributes of the partial identities. Especially for aliases assigned to pseudonyms of other users, *Alias Improvement* may be required for increasing their usability. The improved alias might encode additional knowledge about the communication partner, drawn from additional observations or experiences made in interactions or from knowledge collected over time, respectively. To allow for an easy handling of current aliases and for representing information about former aliases, only pseudonyms are stored as part of application data. Finally, *Alias Replacement* is an especially important component: It has to ensure that only secure pseudonyms leave the client system, i.e., are transferred, and that aliases are represented to users. In order to support local alias management, the *Alias Replacement* is also responsible that the locally-assigned aliases are represented to each user (cf. Fig. 24.2). Finally, replacement has to deal with possible errors resulting from ambiguities and typos during alias/pseudonym mapping and vice versa.
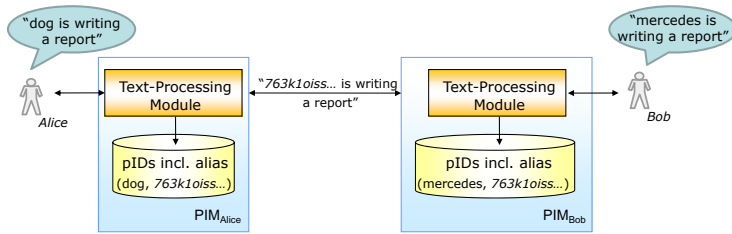
Fig. 24.2 Replacement of aliases and pseudonyms in BluES'n

Realisation in BluES'n

In BluES'n a general pseudonym/alias mapping was integrated which locally assigns an alias to each pseudonym automatically. By means of this approach the user is supported in handling and managing his own pseudonyms as well as pseudonyms of other users in a comprehensible and user-friendly way. Additionally, a first automatic analysis and replacement of pseudonyms appearing in commonly-used content by aliases takes place in communication and cooperation processes. Thus, for example, a replacement within a chat session takes place in case of addressing, i.e., naming, the communication partner directly by means of his locally-assigned alias. The *Alias Replacement* component of BluES'n checks during a user is writing a message whether the typed-in text corresponds to an existing alias/pseudonym mapping. If such a mapping exists, the user is asked on the fly whether the word should be marked as alias. By computing the Levenshtein distance[2], the user is also presented a list of possibly matching aliases. In case of an ambiguity error, he could select the one, he wanted to address.

Supplementing text-based aliases by means of graphical representations for pIDs might raise the recognizability of pIDs of other users. A first approach is described in Sec. 24.4.2. The consideration of additional contextual information for dynamically assigning aliases to pseudonyms are challenging approaches for future work in this field. Additionally, investigations beyond text-based representations should take place in order to support further target audiences, such as acoustic representations for blind people, and user's end devices.

24.4.2 Chernoff Faces

During their work in a complex and collaborative environment such as BluES'n, users will create various pIDs for different scenarios. In order to assess their linkability from the point of view of other users, they should get informed which information others might know, especially, which pID was

---

[2] http://levenshtein.net

used in which scenarios, for which actions, and which information was delivered while working under this pID. Likewise, users should be able to recognise pIDs of other users within a collaborative environment. Consequently, an intuitive and clear representation of pIDs is of fundamental importance. Thereby, a number of different information items must be encoded and represented.

Generally, the representation should enable users (1) to recognise pIDs and (2) to assess their relevant features regarding the current context. It is not possible to consider these two requirements within one single representation: While a static identifier is needed for recognition, the attributes of a pID are context dependent and might also change over time. Therefore, they require a dynamic representation.

Within BluES'n, pictograms in style of Chernoff Faces [Che73] are used to dynamically visualise multivariate data in the shape of a human face [FLBP06]. The alias as static pID identifier is displayed together with the icon. The representation of relevant features of pIDs adheres to fixed rules in order to increase its usability, especially to support users in intuitively recognising these features (Tab. 24.2). Furthermore, the system can automatically generate the corresponding representation and dynamically adapt it.

Table 24.2 Features used for visualisation

Selected feature	Encodes information about
Eyes	Degree of Knowledge
Mouth	Kind of communication
Eyebrows	Links to other pIDs
Shadow behind face	Scope of pID
Color of face	Online state
Margin of face	Active workspace
Additional symbols	Delivery of additional information
Font style of alias	Administrative role

The meaning of the features might differ for own pIDs and pIDs of other users. For example, the degree of knowledge reflects the knowledge others might have about own pIDs. For own pIDs, this value is estimated from the number of actions performed under this pID and the delivered data. In contrast, eyes of pIDs of other users represent the average degree of knowledge the other one might have about the user's own pIDs. Tab. 24.3 shows an example for a pID as well as the representation of a new pID together with an interpretation of the encoded information. Within the current version of BluES'n, only the online state and the alias including the representation of the administrative role are realised. Even this information gives some feedback about pIDs currently working within BluES'n. Future versions will enhance the representation by further features, which is needed for providing privacy-relevant information.

Table 24.3 Visualising relevant features of pIDs within BluES'n

Example pID of other user	• medium degree of knowledge • indirect communication in a functional module occurred • explicit links to other pIDs • pID was only met in one workspace • awareness objects of this pID received; pID is online • subscribed to inactive but opened workspace • notes assigned; pID is tutor in current functional module • pID is participant in active shared workspace
New pID	• least degree of knowledge • no communication • no links to other pIDs

24.4.3   GUI Components: InfoCenter and Echobar

Users do need usable and user-friendly graphical interfaces for getting an overview about their privacy- and group-related awareness information as described above. For that purpose, BluES'n was enhanced by a so-called Info-Center which is a special module being always presented to and accessible to the user. Besides displaying awareness information, it serves also as starting point for configuring several aspects such as the contribution of awareness information or level of partitioning of users' personal data, respectively. By means of three different panels, in the current version of BluES'n awareness information are provided as described in Tab. 24.4.

Table 24.4 Information provided by the BluES'n InfoCenter panels

InfoCenter panel	Information provided by panel
Workspace	The tab provides information about the state of the current workspace and the ongoing activities in this workspace.
About Me	This panel shows information about the user himself and the representation of the user to other members of the current workspace.
Community	All members of the current workspace are listed in this tab. Moreover, detailed information about the participants is available.

In addition, a so-called Echobar was integrated consisting of small traffic lights and an additional status bar which are used to visualise important, privacy-relevant aspects during the work within BluES'n. As soon as an event occurs, the corresponding lamp lights up and an appropriate message is displayed via the status bar.

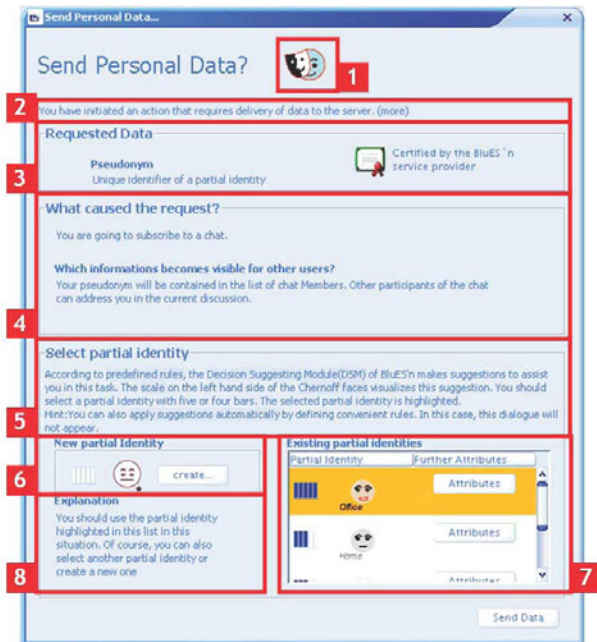### 24.4.4   Adapted "Send Personal Data"-Dialogue

Based on the current context and the user-defined rules, the DSM (cf. Sec. 24.2.2) generates a rating for pIDs already used within BluES'n or suggests generating a new pID, respectively. If an automated decision is not possible, a user dialogue is displayed which has the task to present the rating to the user for supporting him in the decision which pID should be chosen for the next action. PRIME already provides an appropriate dialogue, the so called "Send Personal Data"-Dialogue (cf. Sec. 20.5.1.1). However, this dialogue is tailored for communication with different application providers who request various personal data items from the user. It is not perfectly suited for the use within BluES'n due to the following reasons:

- Section *Select a Template*: The concept of templates shall be used within Blues'n to summarise data that can be requested in specific situations; however, there is no need to select one out of different possible templates as in the scenario considered by the PRIME dialogue.
- Section *Your data*: The PRIME dialogue focuses on representing which data are requested from the user. In BluES'n, information about users arise primarily due to their actions: All actions performed under one pID can be assigned to it and allow one to collect information which might simplify linking different pIDs or linking a pID to a user. A dialogue for BluES'n should instead present the rating of the pIDs already used within BluES'n.
- Section *Will be sent to*: This section informs the user about the communication partner who receives the data. Within BluES'n, this "technical communication partner" is always the BluES'n server. However, other users might recognise the action of a user due to the awareness information. Thus, the user should be informed who might recognise what.
- Section *Purpose*: Only business purpose should be possible; thus, this information could be integrated into the section *Will be sent to*.

To conclude, we integrated an adapted dialogue into BluES'n tailored for the application scenario (Fig. 24.3):

1 A dynamic DSM Configuration Button signals the current settings regarding the context management.
2 Introductory text informing why the dialogue is started.
3 Information about the requested data.
4 Introduction about the information given and handling of this section.

5 Via the "Create"-Button the user can generate a new pID (see "Create new partial Identity"-Dialogue). If the DSM suggests creating a new pID instead of using an already existing one, the button is highlighted orange.

6 The rated list of pIDs – the rating is visualised via the scales on the left hand sides of the Chernoff faces representing the pIDs. If a pID gets the highest rating, it is highlighted. Additionally, the user can have a look at attributes already assigned to a specific pID.

7 Information about the suggestion generated by the DSM.

8 Information about the initiated action and about the degree the provided data might be known to other users.

Fig. 24.3 Adapted "Send Personal Data"-Dialogue (the labelled parts are explained in the text)

24.5 Summary – The Final CeL Prototype

Reducing the collection as well as processing of users' personal data on the services side to a minimum and providing users a possibility to keep track of all transactions of their data are the main objectives of BluES'n. During the

PRIME project, two development cycles were performed resulting in two versions of the application prototype BluES'n: Within the first version, special attention was paid to the integration of basic features of PIM into a comprehensive workspace-based collaborative eLearning prototype. During the realisation of the enhanced second prototype version, requirements on the PRIME integrated prototype were refined, particularly with regard to the support for privacy-enhancements based on PRIME technologies and considering the evaluation results of the previous application prototype version. In particular, revisions and conceptual refinements with respect to access control mechanisms, the realisation of intra-application partitioning and the privacy-aware user interface have been performed. Evaluations of the first prototype have shown that the approaches go in the right direction in terms of concepts, but that concrete implementations lack a systematic usability-engineering process with respect to the application's primary objectives as well as the integration of components of the PRIME toolbox. Thus, beside the development of the 2nd prototype version, the privacy-enhanced eLearning application BluES'n underwent a detailed and comprehensive usability-design process. Comparing both versions, modifications and improvements listed in Tab. 24.5 are part of the final BluES'n version.

According GUI improvements and enhancements of the final prototype version were additionally checked by an internal evaluation focussing especially on usability improvements. Eight students from the field of computer and media science participated as test persons. They had no experiences in using BluES'n; only two of them used other eLearning systems like WebCT before. The evaluation was organised as a collaborative session with two test persons at the same time. Altogether, the evaluation confirmed that the final BluES'n version provides a more usable and user-friendly environment:

- After a short training, based on the additionally-created *BluES'n Getting Started Tutorial*, the system can now be used more intuitively. Furthermore, six of eight test persons indicated that they would potentially use BluES'n in the future. The majority of the test persons stated that they understand and can cope with the PRIME concepts and their integration into BluES'n.
- The *credential handling* in the 1st version was a hard-to-understand and time-consuming process. In contrast, the evaluation of the final prototype version has shown that the revised credential handling, i.e., requesting and granting access to workspaces, is more user-friendly, unobtrusive and comprehensible. The user is no longer confronted with credentials as long strings representations. Now, he can handle them intuitively by understandable dialogues and menus, which was explicitly emphasised by one of the test persons.
- The adapted "Send personal data"-Dialogue provides a more transparent and understandable integration into the application. One test person using

Table 24.5 Summary of enhancements of the BluES'n prototype

Main concepts	Features provided by the enhanced prototype
pIDs, pseudonyms, and aliases	Possibility of defining attributes of users by means of pIDs with BluES'n (currently only pseudonyms as attributes)
	Realisation of pseudonym/alias mapping
Enhancements of access control based on policies and anonymous credentials	Administrative roles and workspace access modes
	Requesting and granting access to shared workspaces
	Simplification/grouping of policies for BluES'n resources
Enhanced context management to realise IAP	Integration of Decision Suggesting Module (DSM)
	Enhanced configuration and monitoring functionalities
Privacy-aware user interface	Revision of GUI and workflows based on evaluation results of version 1 particularly focussing on improving usability
	Provision and representation of group- and privacy-related awareness information
	Realisation of Chernoff faces for visualising pIDs
	Tailoring of integrated PRIME "Send Personal Data"-Dialogue for representing suggestions generated by the DSM and adapting it to the BluES'n specific design
Further concepts and features	Integration of basic reputation management
	Enhancing, respectively adding, communication supporting functional modules *MailForum*, *Chat* and *Group Calendar* as demonstrators of PRIME component's interplay
	Provision of BluES'n Getting Started Tutorial

the highest anonymity level was able to cope with mostly all concepts of pseudonyms and IAP.

- In contrast to the 1st version, information relevant for privacy and group awareness are integrated into and now displayed via specific components in the graphical user interface of BluES'n. Particularly, assigning aliases as user-friendly representation of secure pseudonyms and visualising pIDs by means of Chernoff faces was helpful for working with different pIDs during the evaluation.

Summarising, the architecture of the PRIME toolbox and the provided PRIME Integrated Prototype meet the requirements for establishing a privacy-preserving collaborative working and learning environment for the most part from the perspective of BluES'n. However, the Integrated Prototype is mainly built for bilateral scenarios; consequently, it is not applicable for complex environments such as BluES'n comprising a variety of comprehensive collaboration, cooperation, and communication scenarios. Between interacting users and user groups manifold dynamic and flexible relations exist – so-called *multilateral interactions* (MLI) – resulting in complex requirements concerning

privacy and performance issues which have to be reflected in an appropriate system architecture (Section 22).

24.6   Beyond PRIME – An Outlook

At the end of the project PRIME, with BluES'n a collaborative learning and working environment which is enhanced by privacy functionalities by design was established for the first time. From the functional perspective, the integration of privacy-enhancing components was in the focus – their usable and user-friendly realisation was only a secondary goal. However, for practically-deployable complex and collaborative environments such as BluES'n, the focus has to be changed: Primarily, the user's attention should be on intended learning and working processes, while the management of his privacy preferences and settings comes second – only in the background. Introducing to users the available possibilities for privacy and identity management and offering default settings of possible privacy configurations before using the application for the first time would be an important refinement step. That way, users could concentrate on their actual work at first to become more familiar with the application and thus, would be able to adapt the configurations according to their preferences. In this context, designing concepts for supporting users in making decisions with respect to their desired level of privacy is reasonable.

With respect to collaborative working and learning, the consideration of two aspects is needed: First, the application should be enhanced by functionalities allowing for building up trust despite the possibility for intra-application partitioning of personal data and, thus, an increased level of anonymity. Beside an advanced role concept, a privacy-enhanced reputation system as it is already realised in parts for BluES'n belongs to functionalities addressed in that context. On the other hand, PRIME is primarily focussing on traditional client-server applications especially for service providers and users. Until now, collaborative approaches, where multiple clients respectively their users are interacting with each other, are not considered. However, using privacy-enhancing identity management for use cases on top of such multilateral interactions is thoroughly necessary for performing collaborative as well as privacy-supported team work.

Beyond the mentioned issues, an improvement of performance is necessary for practical deployments. PRIME does not integrate multi-threading technology and, therefore, it runs in just one single thread which means that all requests are sequentially processed, which is not sufficient for a collaborative application using the features of PRIME. Enhancing the collaborative functionalities provided by BluES'n might be also in the focus of further developments. Currently, concepts such as a privacy-enhanced eVoting,

cooperative content creation, and a BluES'n wiki exist only as design ideas or first prototypes. Their integration into BluES'n might contribute to a broader acceptance as well as the applicability of privacy-enhancing collaborative environments in general – and BluES'n in particular.

25

Location-Based Services

Jan Zibuschka[1], Kai Rannenberg[2], and Tobias Kölsch[3]

[1] Fraunhofer IAO
[2] JWG Universität Frankfurt
[3] T-Mobile

25.1 Introduction

Location-based services (LBS) determine the location of the user by using one of several technologies for determining position, and then use the location and other information to provide personalized applications and services. However, a user that employs location-based services on a regular basis faces a potential privacy problem, as location data may be gathered to allow for profiling of the user's movements, which might discern personal information, based on places the user visits regularly or at specific times. Also, the user's personal interests and even health condition may be revealed by the services he uses or his configuration parameters.

Location-based services are employed for a wide range of applications. One popular use case are navigation services, e.g. finding the nearest pharmacy and directing the user there. Typically, users open a connection to a service via their mobile phones, and then the user's position is determined by the mobile operator. The determined position is passed on to the service provider, who compares it to his database. The results, e.g. the 5 nearest pharmacies, are then returned to the user's mobile phone, where they are displayed.

25.2 Privacy in Location-Based Services

In the classic LBS scenario (see Figure 25.1), the only interacting parties are the mobile network operator and the LBS application provider. This may lead
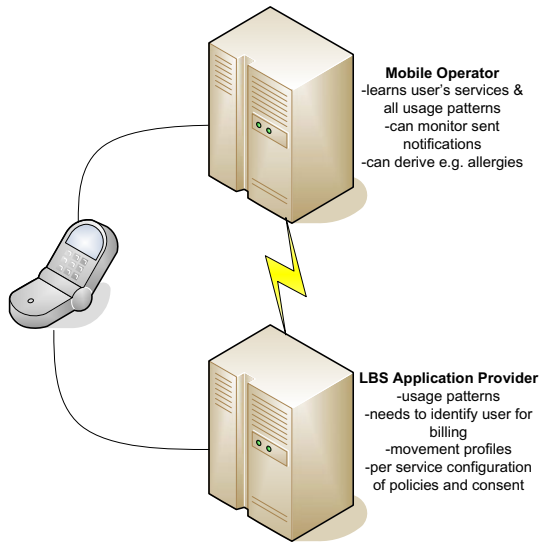
Fig. 25.1 Conventional LBS Deployment

to implementations where the user has to do a lot of configuration and maintenance individually for each service, which makes, for example, configuration of complex privacy policies infeasible from a usability point of view. Additionally, information may be transmitted quite freely between the involved parties, based on the users' agreement to quite general and "generous" privacy policies proposed by LBS application providers.

So the mobile operator usually knows what kind of service the user has accessed, and sees the configuration options transmitted from the user to the service, while the LBS provider would be able to tell which mobile operator the user is using, the user's device identifier (MSISDN) and several contract details, e.g. whether a user may be using a pre-paid card. The service then needs to be customized for usage with a specific mobile operator's location provision interface. Precautions need to be taken to avoid that LBS providers can track users at their discretion. So, LBS are one example for the more general need for solutions that empower users to enforce privacy policies for their personal data, including their location data.

Location information is sensitive data, protected by several privacy regulations. If the data cannot be handled fully anonymously, the user's consent has to be given, as prescribed by the applicable EC directives [Eur95, Eur02], and processed in a comprehensible fashion. Therefore, the handling of such information calls for the integration of identity management components – either for anonymizing the data, or for acquiring the user's consent in a compliant way. Advanced cryptographic protocols like oblivious transfer have been

proposed [KFF$^+$07] for the anonymous rendering of location-based services. However, those implementations are not well-aligned with the common reseller business architecture employed in the mobile scenario. As of such, a proxy-based protocol still has its merits, e.g. in a mobile scenario, when the bandwidth available between mobile operator and location-based service provider is much bigger than the available bandwidth between operator and mobile device, and to support specific business configurations present at many mobile operators. The performance issue may be fixed by running costly obfuscation protocols between mobile operator (who would be aware of the user's location anyway) and service provider, but the business design issues remain.

25.3   Requirements

The requirements on identity management used for location-based services in a mobile network based on [ZFR$^+$07a] have many facets. Investigation of today's LBS scenarios leads to the identification of three main stakeholders, who are present in the vast majority of use cases:

- A *mobile operator* is the owner of the mobile network infrastructure. Its business is to offer the network infrastructure that mobile subscribers use every day, including roaming between different mobile networks. Concerning the provision of location-based services, the mobile operator is often the source for the location information used, and therefore is legally responsible for the release and transfer of the respective data.
- An *LBS application provider* is offering LBSs based on the mobile network infrastructure. A classical example of this are navigation services.
- Last but not least, the *users*, or subscribers, of the services have interests. They are often customers with several contractual relationships: A subscription with the mobile operator enables them to communicate and be mobile, while for specific services they subscribe to the respective specialist service providers.

This section focuses on the business interests of mobile network operators, on regulatory influences in the field of data protection, and on users' privacy requirements in the LBS case. For a more generic discussion of application requirements, see Section 28.

25.3.1   Business Models

The market structure of mobile value-added services, e.g. ring tone downloads or mobile phone logos, is a reseller market, with most offerings coming from third-party providers using interfaces supplied by the mobile network operator. Mobile network operators offer interfaces for infrastructure, identity management and accounting services. Therefore, the requirements of

the involved parties had to be collected, often from different departments, and compiled into a comprehensive requirements document for the prototype (Section 28). The basic architecture of the system should be similar to the structure used in, e.g. ring tone business, allowing for cascading retailers of location-based services, in order to enable external provision of location based services [ZFR+07b]. In addition, implementing a standardized interface has several additional benefits [ZSFR06].

This reseller structure, with a dominant position of the mobile operator, suggests investigating intermediary theory. Efficient organization of selling through service providers, bundled processes, information access and other services can be offered using a unified central interface [ZFR+07a]. The specific advantages of intermediaries concerning, e.g. search time and pricing of traded products have been scientifically investigated. A good overview on general information intermediaries can be found in [SL02] as well as in [Ros99]. Value-added services in the context of mobile network services are particularly suitable for intermediation in several fields:

- Identity management and data protection according to telecommunication laws (see [KZSD07, KZSD08]);
- Bundling of account and risk management services;
- Sale of auxiliary services such as geo-information or usage patterns (see [Fig04]).

Privacy management in the area of conflict between regulation and customer satisfaction is a cost-intensive undertaking, as pointed out by [Pon04]. Therefore, implementing widely-usable, standardized intermediary architectures for identity management is particularly attractive as part of the value creation chain of the cascading location-based services scenario [ZFR+07a].

Another main requirement is the flexibility of the infrastructure component with regard to different national legislations. The users want to employ mobile services regardless of the country they are currently visiting. For voice telephony and data transfer, international roaming already exists. When implementing international infrastructures for location-based services, different local legislations must be considered regarding data protection, data storage, and data monitoring. For the avoidance of new project engineering for each individual country, it is worth to keep uniform intermediate services for privacy management ready and configurable.

25.3.2   Data Protection

On top of the economic requirements presented in Sections 3.1 and 3.2, further requirements result from regulation. Telecommunications in Europe is governed by European Union directives, which are then enforced in the particular countries usually after conversion into national legislation. The Directive 2002/58/EC [Eur02] is relevant for the implementation of location-based services. Legislation on the handling of location data in the context of calls or for

use in, e.g., value-added services is contained in Article 9. For other uses, e.g. for location-based services, data processing must be explained explicitly and a legally effective consent of the data subject must be collected. Moreover, it must be guaranteed that the user is able to temporarily or permanently revoke his or her consent. The homogeneity of the local conversion to national right is unclear in this respect.

Besides, so far there are only a few uniform regulations for the collection of consent, particularly in the case of ad-hoc use of services on mobile devices. From the point of view of users and mobile network operators it is necessary to verify the consent in a reliable manner. As the mobile network operator may become liable for the initial publication of location data, consent needs to be established before the provider of a location-based service receives personal data.

A detailed analysis of the legal requirements of location-based services can be found in [KZSD07] and [KZSD08].

25.4 The Concept of a Location Intermediary

To meet the requirements discussed in Section 25.3, the PRIME LBS application prototype employs an intermediary architecture, introducing an additional party decoupling mobile operator and LBS application provider. This location intermediary offers

- a uniform policy and consent management facility and
- a proxy for users' communications, anonymizing traffic.

So, generally speaking, it decouples the parties involved in the classic scenario (Mobile operator & LBS application provider, as depicted in Figure 25.2) while enabling advanced privacy functionalities in a user-friendly way.

In line with the current market situation, the basic scenario presented here is based on communication between service providers and users that takes place using network infrastructure supplied by mobile operators. Consequently, a very strong position of the mobile network operators with regard to the localization of the users' mobile devices is assumed. Of course, this strong position leads to great responsibility: A network operator releasing customer data to third parties without legal basis of customer's consent is deemed for trouble. In the resulting scenario, a service provider offers location-based services based on its own domain knowledge using users' location data acquired via the mobile network operator. Billing is performed by the mobile network operator, who may charge for access to the users, for necessary localizations, and for offered identity management functions. [ZFR+07a]

The mobile operator performs localization and billing without acquiring any additional personal information of its subscribers (such as service usage or service configuration parameters). Also, the LBS application provider can offer the service without requiring user identification. Thus, the intermediary

can be seen as a kind of identity border (as illustrated in Figure 25.2), hiding the user's identity from the individual LBS application providers.
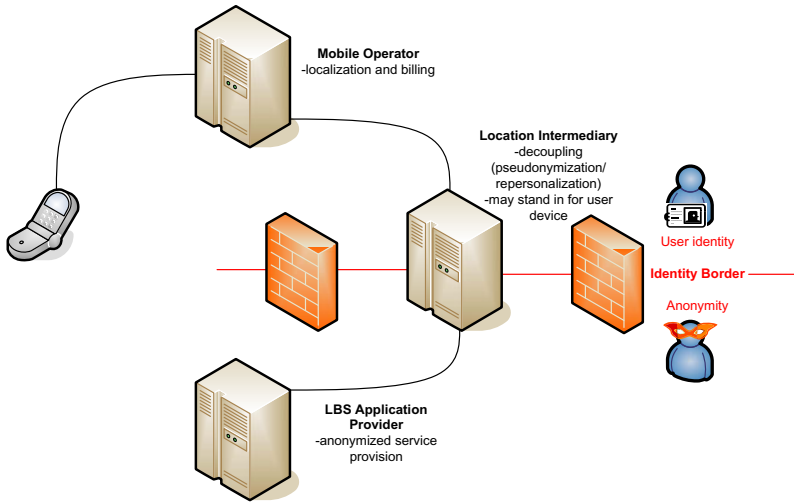
Fig. 25.2 Pseudonymization through Intermediary

The intermediary architecture for LBS offers several key advantages [Ros99, ZFR+07a]:

- *Interoperability:* An intermediary provides a standardized interface for LBS providers, allowing them to access location data in a unified way. This mediation of location information would then allow tapping the network effect immanent in the distributed, multi-party LBS scenario. Mobile operator independence, roaming support, and the unified interface for service providers for easy deployment and migration seem to be viable business propositions in a fast-moving marketplace. Mobility between different services, location sources, involved market players, and applications seems beneficial from users' and service providers' perspectives alike. From an ordinary user's point of view, cost effectiveness, synergy effects, and convenient service usage are major issues.
- *Multi-channel strategy:* An intermediary can collect location data from various sources (GSM, WLAN, and GPS) [AFR05].
- *Synergetic location aggregation:* An intermediary can aggregate multi-channel location information for the benefit of higher quality [LFPR04].
- *Simplification:* Intermediaries simplify process handling for LBS providers by removing the need to negotiate contracts with various location sources.
- *Cross-Operator applications:* Without an intermediary, the creation of user-to-user LBSs with customers using mobile services at distinct mobile operators is much harder.

- *Pricing advantages:* Intermediaries provide many economic benefits in information markets, e.g. an intermediary buys location information from location providers in large amounts, and therefore is in a position to negotiate cheaper prices. For LBSs that consume small quantities of location data, it may be cheaper to acquire location from an intermediary than from a location provider. Other benefits of information intermediaries can be found in [Ros99].

There are different deployment scenarios for the intermediary component, reflecting different business models and organizational structures that are employed in the telecommunications industry. It may be deployed directly at the mobile operator, at a mobile virtual network operator, or outsourced to a completely independent party. This also gives mobile operators the freedom to treat the intermediation of identities as either a core business or as a sideline of the business. In the first case, the intermediation will stay close to the mobile operator but other entities providing a comparable intermediary function will be supported, so that the user has a choice (even if many users practically don't use it). In the second case, the intermediation may simply be outsourced.

Mediating the communication between the different stakeholders (see Figure 25.2), the intermediary offers anonymization of relayed traffic if it is not deployed on the user's device and if some trust can be placed in the entity operating the intermediary. This can act as a fallback solution in cases where the implementation of more elaborate measures (e.g. mixes) is impractical, for example because of restricted client hardware or infrastructure capabilities. However, this will only offer meaningful security guarantees if the connections cannot be eavesdropped at the intermediary by one of the communicating stakeholders. If anonymous communication is available, the intermediary may serve as a rendezvous point for communicating entities [KFKK05]. As already mentioned, advanced cryptographic protocols like oblivious transfer have been proposed [KFF+07] for the privacy-friendly rendering of location-based services. However, in addition to the economic limitations, finding a mechanism that minimizes transferred information in the case of bandwidth-efficient push services is an open research question.

25.5 Prototype Development

The prototype development was conducted in two iterations. The first prototype version demonstrates the feasibility of the approach using the most widely-available technology for a mobile pharmacy search scenario. The second version, a pollen warning service, employs more advanced mobile phones for realizing a more sophisticated privacy-preserving provisioning of location-based push services (where notifications are not triggered by the user). The following sections will present the two prototype versions in more detail, and give an overview of the ensuing commerzialization and deployment at T-Mobile.

After this, an outlook covering the long-term perspectives and lessons learnt from the prototype development will close the chapter.

25.6   PRIME Principles in a Restricted Mobile Environment

Privacy in location-based service scenarios entails several specific design challenges that have been adressed, at least partially, by the LBS application prototype. A short overview will be presented here:

- *Limited device capabilities:* The smaller screen size, combined with limited input devices and hardware performance of mobile devices, makes most user interface designs provided in the context of the PRIME framework not deployable on mobile devices out-of-the-box. The LBS prototype addresses this issue by adapting PRIME user interface concepts where possible (e.g. in the case of the PRIME user-side console, which is displayed in Figure 25.8). In cases where an overly complex user interface on the mobile device would have been necessary, the application prototype resorts to custom solutions tailored towards the scenario, trying to minimize the privacy impact.
- *Visualization/Communication of involved parties:* To make location PETs accessible to the end users and thus deployable in a real-life setting, it was necessary to map the complex server-side deployments to the user interface. An icon for quick identification of the communication partner is provided at the top of all subscription screens (see Figure 25.7).
- *Consent and location disclosure:* A central point of the LBS scenario from the legal point of view is the user's consent when the MO transfers his location information to third-party LBS providers. The implementation has to guarantee that the user's consent has been given under the applying legal parameters (e.g. the application has to make sure that the user has been given enough and correct information to assure his informed consent). This is a requirement for legal compliance of the system, a key requirement to do business in the area of LBS. Adding to the complexity, in the push service scenario, the location disclosure is initiated without user interaction, requiring some means of pre-emptive consent. For this, the user may configure time-based restrictions, which offer a simple yet effective way to restrict access to his location information. The full legal policy is easily available to users, but hidden by default (see Figure 25.7), so it does not obstruct the main workflow. The user is asked for consent in a fashion that should be manageable, yet effective. The concept "giving consent to being tracked" is understood by users.
- *Dynamic personal information:* While classical personal information like name or birth date is relatively stable, location information changes

arbitrarily, and is updated dynamically, without user intervention, in the context of a push service. This poses new challenges to identity management systems, and specifically to the UI, as the prototype tries to conform quite closely to PRIME principles and concepts. For example, the PRIME console will fill up with localization events quite rapidly. To address this, filters to hide periodically updating events were implemented (see Figure 25.8). This also exemplifies the basic principle of adopting PRIME concepts without breaking them: The PRIME Light Console is still very similar to the one used in full PRIME, only adopted for the mobile platform, and augmented by functionality for handling dynamic personal information.

25.7 First Prototype Version

25.7.1 Scenario

John, a travelling salesman, arrives in a city he has visited never before. On arrival, he recognizes his daily medicine is missing. Using his mobile phone, John opens a connection to a pharmacy search service, and his position is determined by the mobile network operator. The determined position is passed on to the pharmacy search service provider who compares it to its database. The results, e.g. the 5 nearest pharmacies, are then returned to John's mobile phone where they are displayed. Obviously, it is important that John's location data are only delivered to a service provider John can trust and only after notifying John and getting his approval.

25.7.2 Implementation

To accommodate the operators' and service providers' interests in a large customer base in the mobile market, a trade-off between large-scale availability of platforms and privacy requirements is necessary, as it is not possible to execute a PRIME client on a standard WAP phone:

On the one hand, the WAP specifications do not necessarily allow for end-to-end encryption and could be intercepted at the WAP-gateway, which is normally under the control of the mobile operator. Also, WAP is sometimes seen as old-fashioned.

On the other hand, WAP is robust and has a high market penetration, as most mobile phones support this standard.

In addition, this enables a smooth migration path to stronger (e.g. Java-based) implementations. The location intermediary in the basic WAP implementation (see Figure 25.4) is responsible for:

- Providing a policy management front end for clients with limited capabilities (e.g. WAP phones, see Figure 25.3);

- Keeping an audit trail for empowering subscribers to trace interactions with certain service providers; and
- Offering pseudonymization and confidentiality of service usage by providing a proxy between mobile operator and service providers.

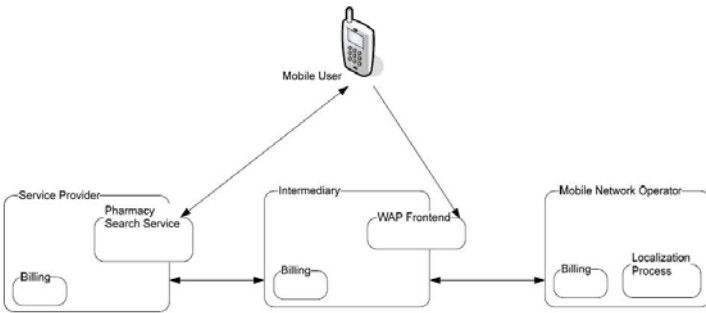**Fig. 25.3** Prototype Version 1 User Interface

Fig. 25.4 Prototype Version 1 Architecture Overview

When a user initiates communications with a service, he is pseudonymized and a communication channel to the MO is established, using the intermediary as a proxy. The relevant privacy policies are checked, and the service can then be rendered. The steps in detail (see Figure 25.4):

- The user contacts the location-based service provider. (Step "search pharmacy" in Figure 25.5)
- The LBS provider requests an access handle for the current global user pseudonym (e.g. IP address, in the case of no anonymous communication infrastructure) via the location intermediary. (Step "request user handle" in Figure 25.5)
- The LBS provider requests user location and payment allocation from the mobile operator. Policies are managed at the location intermediary in the WAP scenario. The mobile operator may then provide user location and a payment handle to the service provider via the intermediary, if a matching policy is available. If no such policy can be found, the system proposes a policy to the user, based on the service's requirements. (Steps till "does a rule exist?" & "recommend rule" in Figure 25.5)
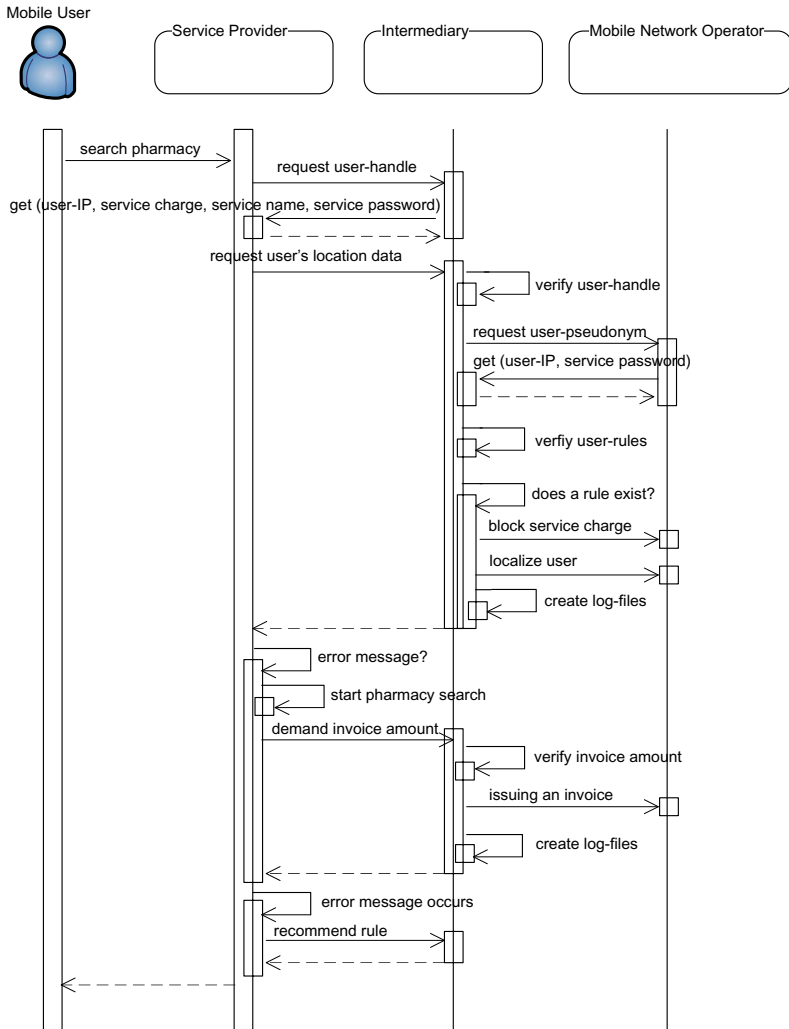
Fig. 25.5 Prototype Version 1 Data Flow

- The LBS provider queries his domain logic, runs the service and provides the result to the user. (Steps up to "start pharmacy search" in Figure 25.5)
- Payment is committed at the mobile operator, again using the intermediary as a pseudonymization proxy. (Invoice-related steps in Figure 25.5)

The implementation was realized to the satisfaction of both project officials and industry partners. It was also used for the development and implementation of a commercial service and product offering as well as for initializing

a roadmap for further privacy-enhancing services, including a second version based on stronger terminals and more sophisticated APIs and protocols.

25.8   Second Prototype Version

25.8.1   Scenario

In the second version, the scenario is a pollen warning service employing more advanced mobile terminals for realizing a sophisticated privacy-preserving provisioning of location-based services, where notifications need not be triggered by the user directly. Once a user is subscribed to the pollen warning service, he will be localized periodically by the mobile operator, and will receive warnings via SMS whenever he enters a pollen area.

25.8.2   Implementation

The implementation of the second prototype version was a two-step process, due to interleaved development of a commercial product based on the first LBS application prototype version (see Section 25.9). This enabled the developers to gather intermediate feedback from the reviewers, and the ability to integrate their valuable comments.

As already mentioned, the second prototype version aimed at supporting push services using state-of-the-art mobile technologies, leveraging the interoperability reached by the architecture already used in the first version. The additional capabilities were applied to the more complex scenario of push services, with a special focus on partitioning of data between the involved entities and the user's control over his personal information.

The prototype workflow can be roughly divided into 3 steps (cf. Figure 25.6):

1. *Service subscription and configuration:* The user first subscribes to a service, configuring his policies and profiles. The configuration of privacy policies (see Figure 25.7) is the most complex interaction in this prototype, and will be discussed in some more detail here. As indicated by the logo at the top of the screen, those will be stored and enforced at the MO (T-Mobile). The location requestor (the LBS application provider) and the charged amount will be displayed, but – at least in the current implementation – cannot be edited by the user, as it reflects the costs for a specific service, which is offered by a specific party and will cost a specific amount of money. However, the system offers the possibility to restrict the localizations to specific times, and is also able to temporarily disable any policy. The user also configures his allergy profile, which will be stored at the intermediary in an obfuscated form.
2. *Determination of region of interest match:* After this, the intermediary regularly receives updates of ROIs and user location, checking whether a user notification is necessary.
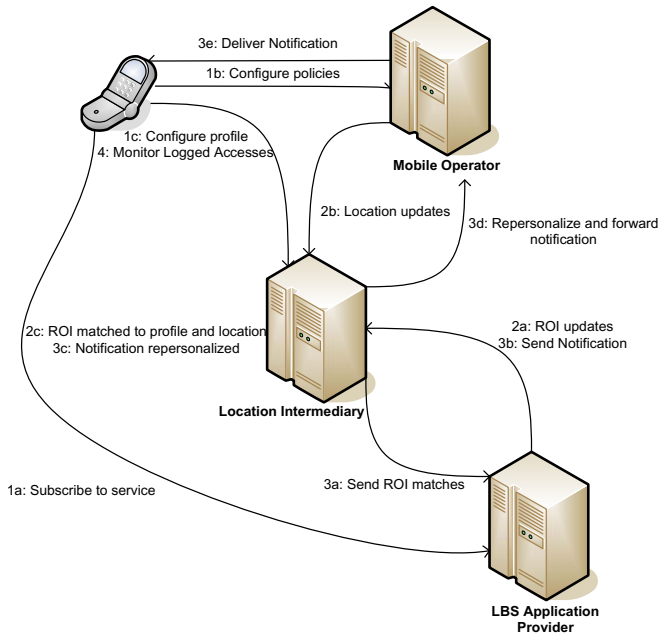
Fig. 25.6 Data Flow
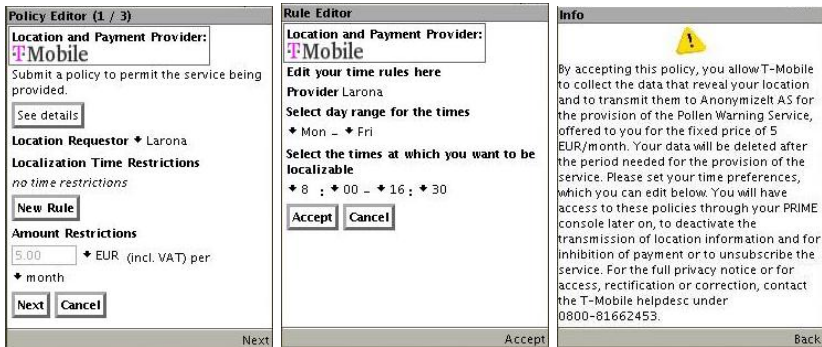
Fig. 25.7 Policy configuration

3. *Notification:* If a match is determined, a notification is prepared by the AP, repersonalized and then transmitted to the user, using, e.g., SMS (see Figure 25.6).
4. *Logging:* In addition to the policy/profile configuration options, the PRIME Light Console also offers a data track offering the user the possibility to audit past transmissions of personal information as well as localizations, policy administration and other relevant activities (see Figure 25.8). There is also an option for quick unsubscription from LBSs.

Fig. 25.8 PRIME light Console

25.9   Commercialization

The LBS prototype presented in the previous sections has been developed by T-Mobile in cooperation with Goethe University Frankfurt. For T-Mobile, the prototype leads to new insights into how privacy-enhanced identity management can be introduced into an 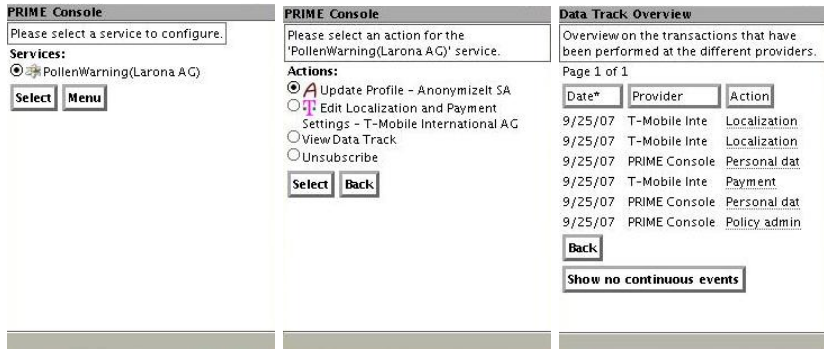m-commerce scenario without restricting the business models. An idea on how privacy enhancing services can be deployed within a telecommunication environment, especially as a standardized identity management system, can enable new and efficient business models in such a scenario. Also, T-Mobile decided to develop a product based on the prototype, which will be covered in some more detail here. An overview of the deployment is given in Figure 25.9.

The product was integrated into the infrastructure using an interceptor pattern: incoming messages are intercepted by a modular access control component, which can be configured by the user using his mobile phone. This enables easy integration and maintenance. Also, the component uses standard-based interfaces, enabling easy interoperability and integration at third parties. Apart from this, however, it is a direct translation of the middleware approach explored during development of the prototypes with its associated flexibility and privacy friendliness.
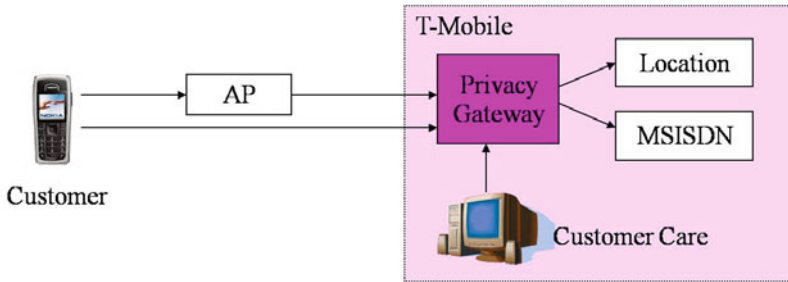
Fig. 25.9 Architecture of commercial product
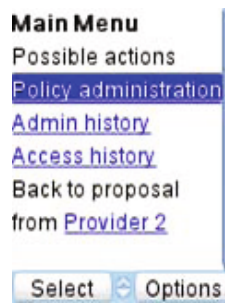
Fig. 25.10 Commercial product user interface

The user interface (see Figure 25.10) is also very similar to what is used in the prototypes, e.g. in the PRIME Light Console.

25.10 Possible Deployment

While intermediary components generally act as middleware, separating the location source from the LBS provider, they don't have to be deployed by independent parties, but may also be deployed as components on the user's device, or on the mobile operator's systems. But, even when independent intermediaries are considered, the central question remains: Which players will step up to take the role of intermediaries? We will briefly evaluate several possible configurations in this section.

1. *Users:* A user may deploy the intermediary on his client. This will minimize the exposure of his location information. However, to ensure tamper resistance, certification of data will have to be done directly by the device. This might require trusted components to make sure that security-critical information is not tampered with. Additionally, the service is more likely to contact the user directly in such a scenario, limiting anonymity.

2. *Mobile Operators:* A mobile operator might want to deploy the intermediary directly at one of its facilities. Most functions of the intermediary, like policy handling, PETs and anonymization towards the service provider may be preserved. Additionally, the mobile operator is already aware of the user's location, so no additional information is spread, and independent, yet potentially trustworthy enough, to certify it. However, as several advanced features of the intermediary, including the anonymous rendezvous functionality, depend on the separation of the participating parties, the organizational structure and potentially employed advanced protocols become a key element for security in this case.

3. *Mobile Virtual Network Operators:* Independent from both mobile operator and LBS service provider, Mobile Virtual Network Operators (MVNOs) deal with customer relations, while a Mobile Operator manages the underlying infrastructure technology. Interpreting identity management as part of customer relations makes a lot of sense, so MVNOs seem to be well-positioned to run location intermediaries, at least when considering organizational structure.

25.11 Outlook

As has been presented in this chapter, location-based services are a promising application area for identity management. However, the mobile scenario also poses several specific challenges in this context. Dynamic personal information needs to be integrated in a system that is usually more tailored towards a database field-like interpretation of identity attributes. We demonstrated a feasible approach by integration of location sources and LBS acess control policies based on them (basic information on the access control component can be found in section 11.4.3). In addition to those special requirements, adressing the mobile space means living on restricted client platforms. Mobile devices have limited capabilites when compared to, e.g., desktop PCs, both with regard to performance and features. This makes a reimplementation of PRIME framework components, and additional customization of UIs, unavoidable. Additionally, the context-rich environment poses a severe performance challenge: there are many services employing personal information, specifically dynamic location information, and a large number of users are subscribed to the infrastructure. Also, proper visualization of location information is not a feature that is usually implemented in identity management systems. Still, the focus on location information in the mobile scenario makes

visualizing user locations on a map, e.g. in context of the PRIME console, or when configuring policies, an attractive solution. Restricted platforms are a challenge here, however. Those points will be taken up again and discussed from a more generic perspective in Section 28.

Beyond a fixed deployment of identity management functionalities at the user or service side, there is also the possibility of a market dominated by independent intermediaries that choose localization and connection options dynamically from a pool of available possibilities – for example, from several MOs and MVNOs – based on the users' policies and preferences. Thus, dynamic party matching recommendations may be used to leverage network effects, building a market that offers ease of development and deployment to service providers while preserving the users' privacy. This raises new requirements for identity management frameworks processing location information, but also presents a promising use case for advanced privacy-respecting features.

26

e-Health

Alberto Sanna, Riccardo Serafin, and Nicola Maganetti

San Raffaele Scientific Institute

26.1 Introduction

In the following chapter we are going to discuss the impact that privacy-enhancing technologies can have on today's healthcare market, which is increasingly focused on the provision of (online) solutions to support the so-called "Continuity of Care". This landscape offers many new opportunities to improve the quality of care, but also poses many new risks with respect to the protection of the individual's identities and privacy.

We will first present the new trends in terms of personalized healthcare and "individual's empowerment", focusing in particular on the domain of self care medication regimes, which constitute a relevant portion of the healthcare market and is a good test case to study the interaction between complex real-life processes and privacy-enhancing technologies.

Within this domain, we will discuss a scenario for privacy-enhanced online drug provision, from drug prescription, to drug preparation and delivery. Afterwards, we will present a proof-of-concept solution that, thanks to PRIME technologies, implements such a scenario while satisfying the seemingly contrasting requirements of providing an integrated service involving several market actors and individual's information exchange among them, and at the same time protecting and enhancing the individual's privacy to the point of allowing the consumer to purchase drugs anonymously.

26.1.1   Definition of "Health" by the World Health Organization (WHO)

In order to properly understand the intimate relationship between healthcare and privacy, it is important to refer to the WHO definition of "Health" from the Constitution of the World Health Organization, July 22, 1948 and related implications as they are unfolding as unequivocal irreversible trends in the medical science and practice. World Health Organization states that: *"Health is a state of complete physical, mental and social wellbeing and not merely the absence of disease or infirmity. The enjoyment of the highest attainable standard of health is one of the fundamental rights of every human being without distinction of race, religion, political belief, economic or social condition"*. Today's healthcare is as close as never in the past to this visionary statement. Illness is no more just an acute event in the individual's life, that must be treated in hospitals where the patient's role is passive and where technological and human resources are focusing their effort to overcome his/her critical health condition. Obviously, this dimension of healthcare is still present, as it will be in the future. In parallel, the key factor that has definitely emerged is the individual's medical and societal need of engaging the person in the process of his/her own care, known as "individual's empowerment". An empowered person is eventually relying on his/her awareness, being able to take informed decisions in everyday life and to pursue personalized disease prevention-oriented[1] medical prescriptions. Individual's empowerment is a critical factor both to achieve changes in individual's behavior, according to the prevention criteria set by the medical science, and to sustain the exponential growth of health demand[2] in a context of scarcely available existing healthcare resources (doctors, nurses, pharmacists, health technologies and products, inpatient/outpatient facilities[3], etc.) and ageing of population. The individual's empowerment model in healthcare implies a continuum of care throughout individual's daily life.

26.1.2   Continuity of Care and Impact on Individual's Life

Continuity of Care is enabled by a seamless connection between individuals and caregivers, and results in a bidirectional information flow between

---

[1] Primary prevention diminishing the risk of a disease occurrence; secondary prevention diminishing the risk of chronic disease to evolve to critical stages, requiring hospitalization or other higher individual and societal costs.

[2] Public and private health market incidence on Gross Domestic Product (GDP) in Europe is 8.9% and in USA is 15.3% [EFP08]

[3] An outpatient is a patient who is not hospitalized overnight but who visits a hospital, clinic, or associated facility for diagnosis or treatment. An inpatient on the other hand is "admitted" to the hospital and stays overnight or for an indeterminate time, usually several days or weeks.

the caregiver and the individual. The caregiver (or any decision support system replicating the caregiver knowledge) receives the amount of data needed to understand individual's conditions and context. Once this information is available, it is the caregiver's turn to provide personalized information and raise individual awareness on appropriate behaviour. It must be emphasized that the Continuity of Care model is unequivocally framed within the ethical imperative of respecting the individual's free will: individual's awareness is, in fact, a key long-term factor for stimulating personal health education and motivation, and it is not intended as a tool to bias, manipulate or force individual's free will.

26.1.3   Health and Lifestyle Management

The implicit pervasiveness of health-related services which is, and will be more and more in the future, affecting personal lives becomes evident when taking into account some facts and figures about lifestyle-related diseases and prevalence in Europe:

- The relationship between diet, physical activity and health has been scientifically established, in particular regarding the role of lifestyles as determinants of chronic non-communicable diseases and conditions such as obesity, heart disease, type 2 diabetes, hypertension, cancer and osteoporosis [Exp03].
- Unhealthy diets and lack of physical activity are the leading causes of avoidable illness and premature death in Europe [Wor02b].
- Although cardiovascular disease and cancer still constitute the principal causes of mortality in Europe, the prevalence of obesity, overweight and type 2 diabetes is increasing in all regions of Europe at unsettling rates.
- The rising rates of obesity across Europe, especially among young people, have alarmed health experts, the media and the population at large, and are a major public health concern. Evidence from population surveys suggests that obesity levels in the EU have risen by between 10-40% over the past decade, and current data suggest that the range of obesity prevalence in EU countries is from 10% to 27% in men and up to 38% in women [Int05].
- In some EU countries more than half the adult population is overweight [Wor02a] (Body Mass Index > 25), and in parts of Europe (Finland, Germany, Greece, Cyprus, the Czech Republic, Slovakia and Malta) the combination of reported overweight and obesity in men exceeds the 67% prevalence found in the USA's most recent survey [Int05].

To complete the overview of lifestyle-related diseases and prevalence it is also important to take into account how younger generations are facing similar, if not worse in perspective, conditions:

- Excess body weight in children is of particular concern as across the EU 14 million children are estimated to be overweight and a further three

million classed as obese. Moreover, the number of overweight children is increasing rapidly, currently rising by 400,000 a year [Rie08].

- Overweight is associated with a number of co-morbidities in children. Although the amount of information available about youth is less than that about adults, it is clear that children experience many detrimental effects of overweight similar to adults.

The epidemiological context described above requires intervention on vast population, potentially all individuals, to be performed in a timely, context-sensitive and personalized way. The intervention model is based on the exchange of detailed data and information during daily life on various health-related topics, for example physical activity, nutrition, physiological data monitoring, therapies, etc. In fact, from an individual's perspective, a self care management process, e.g. in nutrition, implies being seamlessly assisted in taking informed decisions and acting accordingly, to achieve and maintain healthier behaviours when buying and consuming foods (taking into account various nutritional parameters, e.g. ingredients, Recommended Daily Doses of nutrients, calories, vitamins, sodium, fat/saturated fatty acids that have a relevant impact on individual's health). Furthermore, the traditional market segments are blurring, as it is clearly demonstrated in the Food & Beverage sector that is developing an increasing number and variety of consumer products targeting health outcomes and health-specific targets (e.g., functional food/beverages, enriched food/beverages, food/beverages integrators, "Over The Counter" – OTC – products) that are available to the consumers in supermarkets without any specific need of medical advice or prescription.

Presently, the self care patient management model is mainly relying on the wrong assumption that, given specific tasks and assignments to the individual by his/her doctor, he/she will be able to unequivocally understand the conditions under which tasks should be performed and act accordingly, i.e. as per doctor's prescriptions. Unfortunately, such a reliance on the individual's autonomy has already proven to have a significant ratio of failure, thus demanding for innovative solutions and approaches to be developed, all of them implying more invasive approaches in personal life, as it has been described above. A paradigmatic example of such need for improving efficiency and effectiveness of self care regimen is represented by the self care medication regimen.

26.1.4   The Self Care Medication Regimen and the Opportunity for Privacy-Enhanced Processes and Services

The pharmaceutical market represents 16.6% of the healthcare market in Europe and 12.4% in USA [EFP08]. The impact of the pharmaceutical market is of paramount importance both from the economic and the individual well-being perspective. The need to improve efficiency and effectiveness of self care regimen is a paradigmatic example of the entire healthcare sector, and sets the

reference context for privacy-enhanced processes and services. In this perspective, four key aspects are reported in order to give evidence on the specific added value and competitive positioning of processes and services designed and operated under the privacy-enhanced criteria and technologies developed in PRIME. As said before, today's health services are mainly relying on the assumption that, given specific tasks and assignments to the individual, he/she will be able to unequivocally understand conditions under which tasks should be performed and act accordingly, i.e. as prescribed by his/her doctor. On the delivery side, the assumption is that procedures and controls are duly performed in each and every step of the process. Unfortunately, such a reliance on the individual's autonomy and consistence of delivery operations has already proven to have a significant ratio of failure and drawbacks: thus, services and processes must be re-designed and re-engineered in order integrate several presently fragmented processes and to achieve an higher level of patient services in medical as well as in the related delivery services. The criteria and technologies developed in PRIME are of paramount importance in order to achieve an higher level of services, without a reduction in the level of patient privacy.

The four key aspects of the self care medication regimen that will be detailed are:

1. Patient Adherence/Compliance to Drug Prescriptions;
2. Online Pharmacies;
3. Adverse Drug Reaction (ADR) Reporting and Pharmacovigilance;
4. Personal Electronic Health Records (PEHR).

26.1.4.1 Patient Adherence/Compliance to Drug Prescriptions

Efficiency and effectiveness of a self care regimen, in this case related to self care drug therapy management, can be measured according to two dimensions: **adherence** (or compliance) to a medication regimen, i.e. the extent to which patients take medication as prescribed by their healthcare providers, and **persistence**, i.e. the extent to which patients maintain adherence in the long term.

Poor adherence to medication regimens accounts for substantial worsening of disease, death, and increased healthcare costs. In the US it has been found that, of all medication-related hospital admissions, 33 to 69 percent are due to poor medication adherence, with a resulting cost of approximately $ 100 billion a year. The average rates of adherence in clinical trials can be remarkably high, owing to the attention study patients receive and to the selection of patients, yet ever clinical trials report average adherence rates of only 43 to 78 percent among patients receiving treatment for chronic conditions [OB05].

Given the impact of poor adherence to long term therapies, the World Health Organization has published a report, in 2003, calling for action to

tackle this important issue [Sab03]. Barriers to medication adherence are numerous, but include the prescription of complex medication regimen, the treatments of a-symptomatic conditions, the perception of risk, etc. Thus, it is of dramatic evidence the need of increasing the level of patient support service trough the provision of adherence-dedicated services that, by definition, will need to extend the bidirectional exchange of data and information between the patient in his/her daily life and the healthcare service provider through a variety of ICT-based enabling technologies and services, i.e.:

- scheduler engine configured by the caregiver,
- reminders/warnings delivery system to distribute messages and information to the patient on personal terminals (e.g., PDAs, smart phone, IPTV),
- automatic identification system (e.g., barcode, tag RFID, NFC) embedded into the personal terminal or into a different device (e.g., smart cabinet) or the package itself (e.g., smart package and/or blister) to provide real-time feedback to the system,
- a rule-based engine reacting to input, according to a pre-determined set of conditions, sending reminders or warnings to the patient, or to entitled third parties (a relative, the doctor) and logging all related data in a Personal Electronic Healthcare Record,
- a variety of medical-related auxiliary services (e.g., educational, informative),
- Adverse Drug Reaction (ADR) Reporting Systems and Pharmacovigilance (cf. Section 26.1.4.3),
- a variety of commercial-related auxiliary services linked to e-commerce and e-pharmacies (e.g., ordering, delivering, supply chain management).

26.1.4.2   Online Pharmacies

In the recent report *The Counterfeiting Superhighway: The growing threat of online pharmacies*, the European Alliance for Access to Safe Medicine (EAASM) points out that: "...untrained, unsuspecting consumers are vulnerable to the potentially lethal outcomes of buying medicines online. The Counterfeiting Superhighway reveals the scope and repercussions of this dangerous practice through extensive research and examination of over 100 online pharmacies and over 30 commonly purchased prescription-only medicines". Key findings from this report are:

- 62% of medicines purchased online are fake or substandard (including medicines indicated to treat serious conditions such as cardiovascular and respiratory disease, neurological disorders, and mental health conditions).
- 95.6% of online pharmacies researched are operating illegally.
- 94% of web sites do not have a named, verifiable pharmacist.
- Over 90% of web sites supply prescription-only medicines without a prescription.

While almost none of the locally diffused retail pharmacies[4] are providing any Internet-based service to consumers, "...hundreds of Web sites are selling drugs as Internet pharmacies: some of them are legitimate by specific legislation, but many offer products and services that are dangerous. Some take advantage of people desperate for relief by offering "miracle cures" for serious illnesses like cancer. Many offer prescription drugs based on answers to an on-line questionnaire. These sites tell you they will save you the "embarrassment" of talking to your doctor about certain prescription drugs, such as Viagra, or drugs to prevent hair loss, or prompt weight loss. What they do not tell you is that it is dangerous to take a prescription drug without being examined in person and monitored by a health care practitioner to make sure the drug is helping you. Buying drugs from Internet pharmacies that do not provide street address and telephone number may pose serious health risks. You have no way of knowing where these companies are located, where they get their drugs from, what is in their drugs, or how to reach them if there is a problem. If you order from these sites, you may get counterfeit drugs with no active ingredients, drugs with the wrong ingredients, drugs with dangerous additives, or drugs past their expiry date. Even if these drugs do not harm you directly or immediately, your condition may get worse without effective treatment. If you order prescription drugs without being examined and monitored by a health care practitioner, you may be misdiagnosed, and miss the opportunity to get an appropriate treatment that would help you. You may also put yourself at risk for drug interactions, or harmful side effects that a qualified health professional could better foresee..." [Hea05].

Independent of the many reasons why people of all countries go and get information about health over the Internet[5], it's a clear fact that the demand is relevant and constantly raising. Thus, we must find a coherent paradigm to address it and to enhance the intrinsic potential of having a connected individual and consumer, as it will be also described in the following paragraph on Pharmacovigilance.

26.1.4.3 Adverse Drug Reaction (ADR): Reporting and Pharmacovigilance

All new medicines introduced on the market are the result of lengthy, costly and risky research and development conducted by pharmaceutical companies. The latest study released [DG07] in 2007 estimated the average cost of researching and developing a new chemical or biological entity at 1,059 million

---

[4] Local pharmacy business models consider proximity as the key accessibility factor, which used to be true and it is true indeed presently; it's also clear, though, that, from a consumer perspective, accessibility is also a matter of time accessibility, which includes the enabling of asynchronous service relationship, to cope with consumers' logistic needs.

[5] Health-related queries are ranked second (after sex-related queries) in the search engines ranking.

Euro. The chances of new substances becoming a marketable medicine remain relatively small (1/5000-1/10000) and the time needed from patent application to enter the approval process (pre-human/pre-clinical trials, clinical trials up to phase III) is 10 years; administrative procedures (from registration to reimbursement) last between 2 and 3 years. After 20 years patents expire and an extension of up to 5 years can be granted.

It is extremely important to understand the life-cycle of a biopharmaceutical product because, contrary to the public's common belief, all drugs are dangerous. Having the drug approved for the healthcare market doesn't mean that it won't cause problems to single individuals or wide group of patients: such problems may range from minor side effects all the way down to permanent disability, life threats and even death [Kel08]. In fact, in spite of the huge amount of resources and time invested before a pharmaceutical product reaches the market, the complete adverse reaction profile of a drug is not known at the time of approval because of the small sample size (populations that numbers from a few hundred to several thousands), short duration, and limited generalizability of pre-approval clinical trials (most trials exclude the elderly, children, pregnant women, patients with multiple diseases, etc.). Studies' participants may not be representative of the real world and, as a result, only the most common dose-related ADRs are detected in the premarketing phase.

The phase of monitoring a drug from the very moment it enters into the market and it is actually used on the targeted real-world patient population is called *Pharmacovigilance*. Pharmaceutical companies, doctors, pharmacists, nurses, and patients are expected to report to authorities any Adverse Drug Reaction caused by the use of drugs. "...The bottom line of recognizing ADRs is this: Whenever a patient experiences what looks like an exacerbation of an existing condition, or when a patient develops what seems like a new medical problem while being treated for something else, the possibility of an ADR must be added to the differential diagnosis. It just may be the drug!...The overall incidence of ADRs is unknown. However, studies have found that about 8% of emergency department patients are there because of an ADR, and 3% to 8% of hospitalized patients were admitted because of an ADR. [HR89] In fact, about 7 of every 100 patients in the hospital will experience a serious ADR during their stay, and about 3 of every 1000 hospitalized patients may die as a result of an ADR [LPC98]. ..." [Kel08].

"...The majority of reports submitted to [US] Food and Drug Administration come from pharmaceutical companies. However, late or non-reporting of case reports by drug companies, or failure to report any adverse event at all, are major problems...In recent years, the FDA has issued several warning letters to companies, primarily as a result of the late reporting of ADRs ..." [Ahm03].

"...Spontaneous reporting systems are the most common, effective and relatively inexpensive methods used in pharmacovigilance...one of the limitations of spontaneous reports is that, in general, they are poorly documented

and the safety evaluator may need to contact the reporter, either directly or indirectly, through the manufacturer, in order to secure follow up information... Another limitation of spontaneous reporting is that it captures only a small fraction of the adverse events that actually take place. The extent of under-reporting is unknown, and depends on the severity of the adverse event, among other factors. One group of researchers estimated that the US Food and Drug Administration receives reports of less than 1% of serious adverse events, whereas another group gave this estimate as between 8% and 13%..." [DG07].

To strengthen the pharmacovigilance system is a clear advantage for the sake of the individual safety and the overall efficiency and effectiveness of the healthcare sector. The under-reporting factor is hindering the ability of the overall healthcare system to properly address the identification, quantification and subsequent management of drug risks. Among all the other stakeholders, patients, patient associations and patient advocacy groups have an evident interest in pursuing more intensive ADR reporting systems, as the reduction of drug-related risks for patients and consumers depend to a large extent on these sources of information.

A 3-year study on Adverse Drug Reaction reporting by patients in the Netherlands highlights that, although clear differences between ADR reports from patients and reports from healthcare professional exists, the similarities between patient reports and reports from healthcare professionals in most frequently-reported ADRs and most frequently-reported drugs are striking. The conclusion is that patient reporting in spontaneous reporting systems is feasible and that it contributes significantly to a reliable pharmacovigilance [dLvH08].

While the collection of these events directly from the patients could constitute an enormous resource for the pharmaceutical companies as well as the healthcare industry at large, obvious implications in terms of privacy protection and consumer profiling emerge. As such, technological solutions should and could be put in place, as PRIME demonstrates, to contain the issue while at the same time maintaining the service feasibility.

26.1.4.4 Personal Electronic Health Records

Two technological and organizational trends of the last years have recently consolidated in a new category of health applications: Personal Electronic Health Records. Merging the development of electronic health records within health institutions, the transition towards a digital healthcare and a new clinical perspective that is highly focused on the patient (patient centric), in the last year the market has seen the release of several PEHR solutions, two of which – Microsoft HealthVaulth [Mic08] and Google Health [Goo08] – have recently received special attention by the media due to the reputation and reach of the respective service providing companies.

These products (and services) allow the patient to create a personal electronic record of his/her health history, including any encounter with the healthcare systems (hospitalizations, ambulatory visits, etc.), tracks of the drug prescriptions (and related remainder services), records of vital signs self measurements (blood pressure, weight, glycemia, etc.) along with screening and prevention services, diets, physical exercise schedules, etc.

Moreover, these system usually foresee a direct connection with the information systems of the other healthcare actors, such that the record can be automatically fed with the information available, for instance, in hospital information systems or drug providers. As an example, consider the Google Health case: at the present stage this service allows the user to import his medical record from several clinic and medical centers (like the Beth Israel Deaconess Medical Center and the Cleveland Clinic) as well as from online drug stores and pharmacies (like the Longs Drug Stores or the Walgreens Pharmacy); and the list of connected services is increasing daily.

In this scenario, which was considered science fiction only a couple of years ago and is instead now becoming an everyday reality, it is very easy to envision a completely virtual, on-line drug provision process encompassing prescription, preparation and delivery. However, the implications in terms of security and privacy protection of these systems remain to be understood and several warnings have been raised recently [Pow08], presenting Orwellian scenarios where the corporations might have access to the complete medical history of the individuals.

26.1.5   Reference Context for Privacy-Enhanced Process and Service Re-engineering Based on the PRIME Concepts Applied to Self Care Drug Therapy Management

As we have extensively described in the above, the evidence of health as a lifestyle-related factor (e.g., pharmaceutical regimens, nutrition, physical activity, physiological parameters monitoring, personal behaviours) implies the need of an enhanced relationship between the individual, the patient, the consumer (of health-impacting products) and:

- a variety of federated health care delivery services and caregivers,
- a variety of non-health industries and services (e.g., retailers, logistic services, information and education services),
- public and private payers and insurances,
- health authorities.

Such an enhanced relationship is a key factor to achieve personal awareness to sustain behavioural changes in daily life, and promoting prevention and healthier lifestyle at the individual level and at the mass level.

Due to the high medical and economical impact of pharmaceutical products and the self care model being extremely diffused, we have selected the self care drug therapy management as the paradigmatic example and we have

analysed the impact of privacy-enhanced process and services re-engineering based on the PRIME concepts of:

- Data minimization,
- Multiple identities,
- User control and consent,
- Privacy policy enforcement,
- Accountability.

The key objective of this approach is to demonstrate feasibility and added value (in terms of potential increased clinical outcomes, increased service provision, increased privacy management) delivered to the individual, the patient and the consumer by heterogeneous federated services cooperating in privacy-enhanced workflows. The next section will describe a proof-of-concept demonstrator that has been realized in order to experiment with this approach and the PRIME technologies.

26.2 A Healthcare Demonstrator: Objectives and Scenario

26.2.1 Objectives

Considering the facts and analysis reported above, within the PRIME project we have decided to develop a proof-of-concept demonstrator that could allow us to show, first-hand and from a pragmatic perspective, how the PRIME technologies could affect an existing healthcare process, creating the premises for new services, based on a customary process but offered in a privacy-enhanced environment, where PETs become the diversifying factor able to provide added value to the healthcare consumer. Following the analysis provided in the previous section, we have focused our demonstrator only on the provision of drugs, from drug prescription to (home) delivery of the same, as such a process allows us to study a real-life healthcare process with strong privacy implications in a service environment that goes beyond a single healthcare provider and that, as such, includes many interesting challenges in terms of privacy protection. The other aspects described above, namely adherence/compliance monitoring and support, ADR reporting and pharmacovigilance can be addressed in a similar fashion, analyzing the existing process and identifying the weak links from a security/privacy perspective.

Consequently, the main objective of our healthcare application prototype is to allow the end user to manage his drug prescription process while reducing the disclosure of his identity information. It encompasses the whole process of drug provision, possibly allowing for a demonstration of how PETs provided by PRIME could help the user in maintaining the desired level of privacy with respect to his/her personal health information while at the same time allowing for retaining the same quality in terms of comfort and service that and on-line drug purchase service could provide.

26.2.2 Scenario

In this section we present the envisioned scenario from the user perspective, highlighting the interactions, the personal data exchanged between the actors and which part of the application needs which personal data.

As mentioned above, our exemplary scenario is related to the management of a drug prescription process, focused on safeguarding the buyer's identity during the purchase transaction while at the same time guaranteeing the legitimacy of such transaction, i.e. that the buyer is rightfully entitled to purchase a given drug. This transaction mainly happens between the buyer, the patient, and healthcare provider, in this case the pharmacy. However, to complete the transaction, other actors should or could be involved in the process: for instance the doctors issuing the prescription, a front-line payer, like an insurance or the social security service, guaranteeing the economical coverage, or a carrier, taking care of proper delivery of the drugs to the patient home.

Summarizing, the following will be the actors of our exemplary scenario:

- John, our patient, who is sick and refers to his doctor for a possible therapy.
- John's doctor, who will assess John's health status, elaborate a diagnosis and prescribe him some drugs; the doctor will therefore have complete access to John's clinical record.
- A pharmacist, who will process John's prescription and prepare a package containing all prescribed medications. The pharmacist may also be involved in more complex operations with respect to picking the right pills, like preparing solutions etc. To do this job, although he doesn't need to know the identity of the patient to which the prescription belongs, he may need to access John's clinical record. However, the access should be restricted to those information items that are relevant to the current prescription and should be explicitly authorized by the patient;
- The pharmacy clerk, who will give John the package containing his medication when he goes to the pharmacy to retrieve them; the clerk will only need to authenticate John and to identify the package containing his drugs;
- Optionally, a carrier that the pharmacy can use to ship the drug package directly to John's house. The carrier will only need to access the address information linked with the package;
- Optionally, a relative of John who could fetch the drug package from the pharmacy if John is not able to do so. This person should have been authorized by John to fetch the drugs on his behalf and he shouldn't have any access to John's health data.

The scenario could of course also be easily extended to other actors, for instance a paying institution which is responsible of providing the economical coverage for John's purchase. This institution may cover several roles: it could be John's private insurance, it could the social security service for countries

where this institution is responsible for this service, or it could even simply be John's credit card provider if John is paying by himself through such a means. In all cases, for the data minimization principle, such an institution should not need to know which drugs John is buying, only that he is entitled to receive those drugs.

In this scenario John is a PRIME user who, from home, has complete access to his clinical information as he is a subscriber of a special personal electronic health record service offered by his doctor and related healthcare institution, a service that he can access by means of his laptop. Moreover, from his house he can decide which doctor he will go to, to be visited and enable him to view all his sensitive personal data. Usually he is followed by his doctor, who periodically visits John and updates his therapy, which is quite complex, comprising several drugs. A very strong trust relationship exists between John and his doctor, and John has granted his doctor complete access to his clinical profile.

Whenever John's doctor creates a new drug prescription for John, it is saved digitally and a pseudonymous identifier is created, a token which can be exchanged with the other actors under the control of the PRIME platform. PRIME technology will ensure that no information about the prescription is leaked to non-authorized parties.

Once the prescription has been created, John or his doctor can decide to use a service offered by some of the pharmacies in John's town: provide the prescription in digital format, have it prepared and packaged by the pharmacy and then fetched by John without having to stand in a queue. In this way, John is assured that when he'll go to the pharmacy all his medications will be available and that he'll not have to waste time there. Moreover, the service ensures maximum privacy guaranteeing that no one at the pharmacy will be able to link John with his medications, possibly disclosing sensitive information regarding his health status.

In order to achieve this result, the token mentioned before is electronically "shown" to the pharmacy for processing. In the pharmacy, a pharmacist will retrieve the list of drugs associated with this token, in a first stage without accessing any other information, not even the dosage. However, if the pharmacist needs to prepare some particular drug, for instance a solution, or if he suspects that the prescription may be wrong, for example two highly interacting drugs have been prescribed, he can ask to access more information about the prescription and possibly the clinical profile of the patient. If this happens, John will be notified of the request and will be able to decide if he wishes to authorize the pharmacist to access that information. In any case, John's identification details will never be disclosed to the pharmacist. Once the pharmacist is satisfied, he will then prepare a package containing all the prescribed drugs. The package will be prepared in such a way that it will not be possible to identify its content and no information identifying John will be attached to it. The only identification information will be a newly created code that will be digitally linked with the prescription token. A tag, maybe

RFID, will be then printed and attached to the package. Finally, the package will be moved to the pharmacy front-end and John will be notified that his prescription is ready for pick up.

John will therefore go to the pharmacy and will ask the clerk for getting his drugs. John provides a pseudonymous identity token to the clerk. The clerk will retrieve the list of available tokens belonging to John (there may be more than one prescription pending) based on the token shown by John. For each token he will retrieve the package identification code and, after retrieving the correct package, he'll give it to John. The payment can be handled by any means, using cash being the best for preserving the leakage of personal information. The clerk, of course, should not learn John's identity.

Optionally, John may ask for an added service: he can decide to have the package sent to his house by express courier. If John opts for this option, the carrier is notified of a pick-up at the pharmacy location with the list of all packages to be collected. Once the pick-up is completed, the carrier will be able to retrieve the package destination via the PRIME platform. Of course, the carrier will not know the content of the package (apart from being drugs) but will be able to access John's name and address, that is, the only information required for delivering the package(s) to John's home.

Another option is for John to delegate the pick-up of the package to one of his relatives or any other trusted person. In this case, he could authorize the intended person for the pick-up by providing him an electronic credential. Once this person goes to the pharmacy, she will authenticate herself, the clerk will be able to identify the package, or prescription token, which she has been authorized for and hand the package over to her. John will be notified that the pickup happened.

26.2.3   Collaboration with Other European Research Initiatives

The demonstrator has been developed in collaboration with another European Research project within the 6th Framework Programme: PIPS – Personalized Information Platform for life and health Services [PIP04], which aims at creating novel healthcare delivery models by building an environment for Health and Knowledge Services Support.

This environment integrates different technologies in order to enable healthcare professionals to get access to relevant, updated medical knowledge, and European citizens to choose healthier lifestyles. The project aims at bringing together healthcare suppliers, individuals, public organizations, food/drug industry and services, researchers, and health related policy makers in order to create a dynamic knowledge environment. This dynamic environment builds on traditional and new approaches for handling knowledge from current medical practice, evidence-based medicine, and disparate knowledge sources from health/nutrition domains.

The philosophy underlying the PIPS project is to provide an integrated environment that enables the interaction of different types of users with

conventional computers as well as small and ubiquitous devices, such as mobile phones, and medical devices, with the aim of providing them with personalized advice. The PIPS platform combines a number of technologies in order to generate personalized advice, such as software agents, intelligent decision making, natural language generation, and knowledge management.

In the PIPS project, major attention is dedicated to the issue of promoting compliance to the medical advice. The PIPS philosophy, in accordance with recent research in health promotion, is that the patient or healthy person has to have the locus of control of his own behaviour, in order for the advice to be completely understood and put into practice.

In particular, the PIPS project focused one of its scenarios in the context of drug management and support to drug therapy compliance. The project has developed a portal, much like the PEHR mentioned above, that allows the doctor to fill in a drug prescription and the user to access information about such prescription along with additional services: for instance the system generates reminders for the therapy or the patient can access a specialized drug cabinet, to be put in the patient's home, able to recognize the drugs extracted by the user (thanks to RFID technology) and compare them with the prescribed therapy, providing real-time alerts for uncompliances, errors, interactions, etc. The platform developed by the PIPS project has been used as the starting point for the PRIME healthcare demonstrator, covering the roles of PEHR and the doctor institutional health record components as defined in Section 26.4.1. For the other components mockup systems have been created just to demonstrate how the complete flow could work and experiment with the usage of the PRIME toolbox.

26.3 Application Requirements

As mentioned above, the primary objective of the healthcare demonstrator is to define a solution that shall enable

- the consumer, i.e. the patient, to buy drugs anonymously over the internet without releasing undesired personal information;
- the provider, i.e. the pharmacy, to provide drugs anonymously over the internet while guaranteeing that requirements of ethics and law are respected, meaning that only patients who are entitled to received a certain drug, for instance due to a prescription by a doctor, actually receive such a drug.

From this perspective, the proposed solution should allow

1. doctors to create a drug prescription and provide an electronic signature that certifies the link between the patient and the prescribed drugs. Doctors will also specify the drug therapy, i.e. the posology, when the drugs should be taken and any other indication the patient should follow while

assuming the prescribed drugs. These additional information items, like the daily quantity, are needed by the patient to correctly enact his therapy but are in principle not needed by the other actors involved in the process;

2. the patient to obtain such a prescription, understand it and use it to purchase the drugs he is entitled to with the freedom of selecting a specific drug provider. At the same time, the patient should be empowered to not disclose his personal identifying information to the supplier and furthermore to provide the supplier with the smallest set of information needed to complete the transaction, which is only the name of the drugs to be purchased;

3. the patient to request a home delivery of his drugs, without revealing his address to the drug supplier or the drugs he is purchasing to the carrier;

4. the pharmacy to obtain proof that the drugs have been actually prescribed to the requesting individual by an entitled doctor;

5. the pharmacy to obtain proof that the person they are delivering the drugs to, directly or indirectly, is the entitled one either directly, i.e. is the patient, or indirectly, i.e. is a patient delegate selected by the patient himself. This proof, however, should not require the patient to access any, possibly untrusted, system present in the pharmacy to provide his identification information as well as require an (online) interaction between a trusted personal patient device and the pharmacy system for this identification to happen (as it will impose infrastructure requirements difficult to enforce in all pharmacy front-offices).

Moreover, it should be guaranteed that the patient retains control over all the information he is releasing during the transaction. As such, the system should allow:

1. The patient to know what information has been released, directly or indirectly, to whom, when and for which purpose. This will allow the patient to request, for instance, the deletion of such information;

2. The patient to be sure that the released information will be kept and used only for the specified purposes and that after legal requirements have been fulfilled, those information will be deleted by the receiving parties.

Finally, although each actor alone should not be able to retrieve all information about the transaction, the authorities should be able to reconstruct each aspect of it such that, in case any problem should arise, responsibility could be determined and faulting parties accounted for.

The next section will explain how the PETs developed by the PRIME project could be used to satisfy these requirements in our exemplary scenario.

26.4   Application Demonstrator Architecture

To satisfy the first set of requirements described in the previous section, the anonymous credential system offered by PRIME can be used, along with a proper design of the process. To describe the solution, we will first present the main components of the architecture and then proceed with the details of the flow of information between the actors and the components.
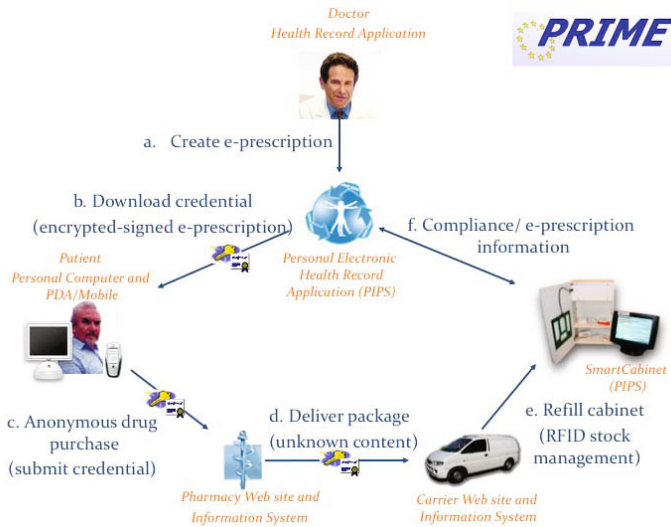
26.4.1   Demonstrator Components

Our exemplary system is composed by the following main components:

- The doctor's institutional health record, which is used to generate the drug prescription and store the patient medical record at the healthcare institution site. This system will be able to communicate with the next component exchanging information about, among other things, drug prescriptions.
- The Personal Electronic Health Record employed by the user to manage his health information.
- The personal computer of the patient with its browser and the PRIME system, with the console, installed on it. The system is used by the patient both to access his PEHR and to access the pharmacy web site to create the purchase order;
- A portable trusted personal device, like a mobile phone or PDA, that the user can carry with him and that could display content used to pseudonymously identify and authenticate the user;
- The pharmacy web site and related electronic management system, which is used to receive orders on line, as well as accessed by the pharmacists and the clerks to prepare and deliver those orders;
- The carrier web site and related electronic management system, which is used to receive delivery orders and pickup information, as well as accessed by the carrier personnel to complete the delivery.
- The SmartCabinet, a common drug cabinet equipped with RFID technology that is able to detect patient interaction with the drug packages, track compliance information as well as display current prescriptions, information leaflets, personalized warnings, etc.

26.4.2   Privacy-Enhanced Online Drug Purchase: Information Flow

A summary of a possible flow between these components is described in Figure 26.1, while Figures 26.2 through 26.4 illustrates in detail the process, which can be described as follows:

1. The doctor uses his institutional health record application to generate and store the prescription, which is then imported into the user's PEHR at his request.

Fig. 26.1 Healthcare demonstrator components and information flow overview

2. Once at home, the patient connects to his PEHR and requests the import of his drug prescription from the doctor's health management system. During this process, a credential for the user is created using the idemix technology. It contains the list of drugs the user should be taking, the posology, indications, etc. along with an electronic signature by the doctor that certifies the validity of the prescription[6]. The PEHR application stores the credential for subsequent download by the patient and uses the information contained within to fill the drug prescription details in the patient records.

3. Once the process is completed, the patient downloads the newly generated single-show credential into his PRIME system, such that it will be available afterwards for the actual purchase at the pharmacy. Along with the credential, the user also downloads a unique random image which will later be used to identify and authenticate him. The same image is also sent to his mobile phone or PDA.

4. The patient selects an online pharmacy store and accesses its web site. He has the option to create a new, pseudonymous, identity to access the site or to reuse a previously-created one if he wishes to use premium services that exploit the knowledge about previous transactions. It is up to the patient to decide if he wishes to maintain maximum privacy by creating

---

[6] The details of the signature process will depends on PKI in place in the given environment. In any case, it is assumed that the pharmacy will have the means to verify the validity and authenticity of this signature.
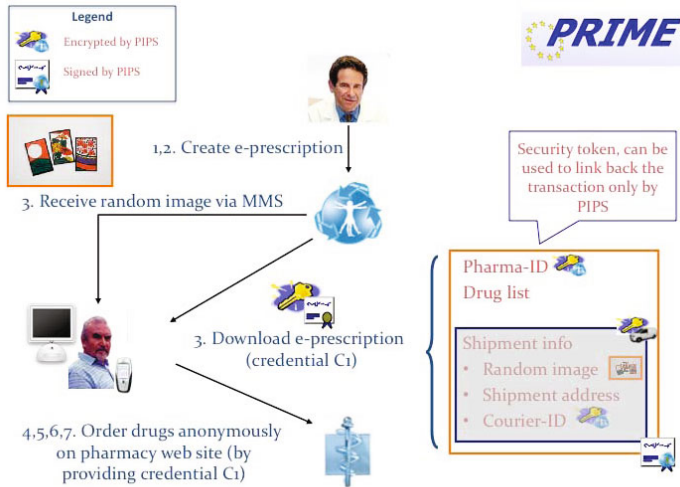
Fig. 26.2 Healthcare demonstrator details: Steps 1-7

a new, totally unlikable, identity or reuse an existing one and, as such, augment the risk of privacy exposure.

5. The patient then accesses the drug purchase page. The resource is protected by PRIME technology which requires, through its access control policy, the user to provide the aforementioned credential to access the page. Once the credential is successfully shown, through the PRIME protocol run between the user-side and services-side systems, the pharmacy web site analyzes the received data (i.e. the list of drugs, the authentication image) and dynamically generates a web page with the list of drugs to be purchased, prices, options for the packages, etc.

   It is important to note that, thanks to the private credential feature provided by idemix, only a portion of the original credential is provided to the pharmacy leaving out, for instance, the posology and additional indications which are needed by the patient to correctly follow his therapy but that are not needed to prepare the prescription and could, instead, provide additional unwanted indication about the patient's disease. At the same time, thanks to the credential show protocol of idemix, the validity of the partially-revealed information of the credential can be guaranteed.

   In the show protocol for the credential, the pharmacy web site's PRIME system verifies the correctness of the digital signature of the credential through a zero-knowledge protocol in order to ascertain the patient's eligibility to obtain the selected drugs.

6. The patient is now able to complete the ordering process, where he will also have the option to select a direct pickup at the pharmacy site or home

delivery, in which case he must select a carrier and provide his address details. These information items, if provided, are sent to the pharmacy web site encrypted with the carrier public key, to ensure non-disclosure towards the pharmacy itself.

7. The information about the drug order is stored in the pharmacy's information system.

8. In the pharmacy back-office a pharmacist accesses the order record, retrieves the list of drugs and prepares a package with the desired items. The package is sealed and provided with a unique identifier which is associated to the order. If needed, an order for the package pickup is sent to the carrier information system including the package reference number and the encrypted address details provided by the user along with the authentication image. At this point, the package is either provided to the pharmacy front-office or picked up by the carrier.

9. Optionally, when the package preparation has been completed the pharmacy information system could send a notice to the patient, informing him that the package is ready and providing details about the package content, like the identification codes of the drugs included. If the PEHR of the user is known, these information could be sent directly from the pharmacy information system to the PEHR, of course under the assumption that the PEHR, being a trusted application, is able to resolve the pseudonymous identifier used by the patient for the transaction back the PEHR identity. This can be realized, as depicted in Figures 26.2 and 26.3 by including a security token that points to the PEHR identity, but that can be resolved only by the PEHR system itself, in the credential used to generate the order.

10. At this point the patient should either pick the package up in person at the pharmacy front office or receive the package at home.

In the first case, the patient will go to the pharmacy and provide the clerk with the order reference number. This will allow the clerk to retrieve *(a)* the desired package and *(b)* the authentication image associated with the order. The patient will therefore be requested to display his own copy of the image on his mobile phone, allowing the clerk to verify the matching and ensure that the package is delivered to the right person. In this way, the link between the patient and the package, represented by the electronic prescription, can be verified without requiring the release of the patient identity. If the patient is unable to pick up the package himself, he can decide to delegate this task to another person, by sending him, for instance by MMS, the order reference number and the authentication image.[7]

---

[7] The mechanism of an authentication image has been chosen for compatibility with most modern mobile phones and its simple delegatability. Though, the binding to the person is not as strong as one could achieve using credential technology. Using features of anonymous credential systems and the assumption that the protocols were implemented also on the mobile device, one could obtain a stronger system than the one explained.
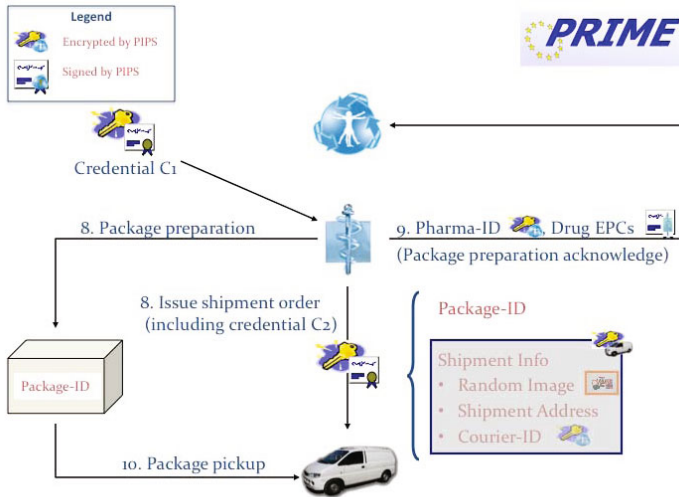
Fig. 26.3 Healthcare demonstrator details: Steps 8-10

In the second case, i.e. the patient has arranged a home delivery, the carrier will pick up the package directly at the pharmacy back-office and, via the package reference number, will be able to retrieve the address information from his information system and deliver the package. At delivery, the carrier operator will follow the same authentication procedure based on the shared image to ensure that the package is delivered to the right person.

11. Finally the patient can refill his drug cabinet with the drugs just received. If enabled, during this operation a check can be made verifying that the drugs received corresponds both to the prescription obtained from the doctor and to the ones delivered by the pharmacy (i.e., no tampering has occurred during the shipping). When the patient will actually take the drugs, the cabinet will fill the PEHR providing both to the patient and his doctor useful information about therapy compliance.

26.4.3 Data Track and Obligations: Ensuring User Control

The second set of requirements, namely guaranteeing that the user is always in control of his personal information, can be obtained by employing two other technologies provided by the PRIME toolbox: the data track and the obligation systems. We start from the latter as it could be employed also to enhance the functionalities of the former in the context of Business-to-Business (B2B) service chains.
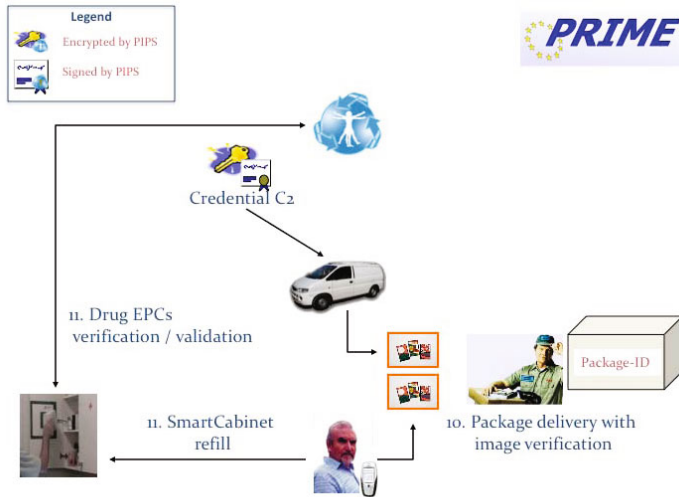
Fig. 26.4 Healthcare demonstrator details: Steps 10-11

Thanks to the obligation framework offered by the PRIME toolbox, it is possible to attach sticky policies to the released information and be sure that a PRIME-compliant service provider will abide to those policies: the policies can be used for several purposes, like ensuring that personal information is retained only for the purpose of the transaction and deleted afterwards (or in compliance with legal requirements). For instance, the pharmacy could be requested to retain information about the order for a certain number of years by the local regulation, but the obligation systems could ensure that the information is deleted from the pharmacy information systems after such a retention period. On the other hand, the information about the delivery address could be deleted immediately after the delivery has been completed from the carrier information system.

Moreover, obligations can be used to enforce purpose binding and opt-in/opt-out clauses for secondary uses. For instance, the PRIME infrastructure could verify and enforce that access to the transaction information is allowed to the pharmacy information system only for the purpose of managing the transaction itself and not, for instance, to generate consumption reports for the pharmaceutical companies.

Obligations could also be used to realize that when a system in the scenario receives personal information about him, he gets notified about this, such that he can be aware of who has received his information, when and why. Functionality related to this is already included in the Data Track feature offered by the PRIME system. The Data Track allows the user to keep the history of the data exchanged with other parties, but currently it is limited

to the transactions directly between the user and the other party. In case of B2B communications involving user data, for instance, the transfer of the address details between the pharmacy and the carrier, the Data Track history is not updated in the current implementation. However, through obligations it could be possible to enforce a policy specifying that, upon receiving the data, the receiving party should notify the data subject to allow for the update of the user Data Track history. Such an update by the recipient would, in the scenario of the user being anonymous, require an anonymuos callback channel to the user.

The Data Track feature of the PRIME toolbox offers also another important functionality: being a collector of the history of all user transactions, the Data Track plays, together with transaction records of other parties involved in the interactions, a fundamental role in the audit processes. It can, in fact, be used by the authorities and under regulated circumstances, to trace all exchanges belonging to a transaction and to link the records present at the different institutions involved in said transaction. For instance, in case of the wrong delivery of a drug, which could eventually lead to serious patient injury and death, the authority could access the Data Track history and retrieve the identities and transaction identifiers (like the order reference number, the package reference number, etc.) used for each part of the transaction. With that information it could be possible to determine the responsible party in case of a mistake in the drug prescription, or in the package preparation or even in the delivery (if certain particular environmental conditions should have been guaranteed).

26.5 Conclusion

The key factor that has definitely emerged in the healthcare domain is the need of engaging the individual in the process of his/her own care, known as "individual's empowerment". The medical science has clearly proven the need to shift to a preventive and personalized medical approach, as lifestyle and personal behaviors are the building blocks of the individual's well-being.

To achieve the preventive and personalized medical approach, it is necessary to aggregate a vast network of competences and services, both in the medical domain (different and specialized healthcare providers and caregivers) and in the consumer domain (value chains of lifestyle-related consumer products and services).

In order to deliver efficient and effective services to the individual, it is necessary to provide a coordinated set of services among the different actors. In order to federate a multiplicity of processes and services, it is necessary that PRIME criteria are used in the phase of federated service design and process re-engineering, to avoid that higher levels of healthcare services are achieved at the expense of the consumer's privacy, to enable innovative services and

to generate an information space that can be used for the evolution of life sciences.

Due to the high economic, social, scientific and medical impact of the pharmaceutical domain (including: manufacturers, distributors, private and public insurance, healthcare providers, medical/nursing/pharmaceutical professional associations, patient associations and patient advocacy groups), self care drug therapy management offers the highest acceptance potential, and the highest probability to demonstrate the return on investment.

In this perspective, positive results have been experienced in a practical and pragmatic feasibility study within the PRIME project on online drug provision. Although the specificity of the scope/objective described, the approach has proven compatible in supporting other relevant scopes/objectives in self care drug therapy management as, for example, adherence/compliance improvement, Adverse Drug Reaction reporting and integration in Personal Electronic Health Records. Thus, in order to scale up the visibility and the market value of PRIME, it will be necessary to extend the scope of a feasibility study to all of the four dimensions of the self care drug therapy management mentioned above, and running a pilot with the aim of disseminating the evidence of adding value to a relevant healthcare market segment by means of privacy-enhancing technologies.

Another relevant area of opportunity for privacy-enhancing technology in healthcare reached a consensus among participants (senior executives responsible for their organization's strategic IT, e-Health, health promotion, and disease management vision for health plans, self insured employers and Federal and State payers) at the Payer Executive Summit, Washington, Dec. 10, 2007, a session of the World Congress of Innovation and Technology in Healthcare [SS07].

In the context of a competitive and private healthcare market, as it is the US healthcare market and as European trends are demonstrating it is rapidly emerging to complement public healthcare, privacy-enhanced healthcare services may play an extremely relevant role as an asset to position the players and providers in the most profitable high-end segment of the market and as a key competitive and differential advantage for customer satisfaction.

In conclusion, the potential of privacy-enhancing technology is extremely high in the healthcare market, provided it is properly positioned in the healthcare value chain and that it could prove its technological maturity with pragmatic approaches, i.e., targeted feasibility studies and pilots demonstrating the ability to manage privacy added value generation in real re-engineered processes.

Airport Security Controls

Prototype Summary

Ioannis Vakalis

Joint Research Centre

27.1 Introduction

The increase in transport security awareness after the 11th of Sept. 2001 has led to safety measures and security checks on travellers associated with distress and delays for airline passengers. In addition, these measures are generally uncoordinated and strongly vary worldwide according to the authority in charge for the security controls. The responsible authority may be the airport (most of the times under police), the police or, in a few cases, the operating airline.

Faced with the problem of long queues, it became accepted that people recognised as "trusted travellers" may be subject to less hindering and delaying controls. "Trusted traveller schemes" are now being implemented. These schemes provide access to a quicker "green line" through security controls for those passengers having provided information establishing their trustworthiness. A precondition for this is a positive identification of the trusted person, i.e. that the physical person seeking access is indeed the "trusted person". This implies the use of biometrics.

The Airport Security Controls prototype is an experimental implementation of such a scheme to investigate the privacy and other related issues rising with the use of personal and biometric information. This prototype actually simulates the stages of the airport check-in boarding process. This pilot implementation was built at the JRC in a SERAC Unit laboratory in the framework of the PRIME project.

In this chapter we concisely describe the ASC [PRI04, VW04, VR05] or "Trusted Traveller" prototype. The aim is to point out the most important aspects of this prototype and the implications that the application of these technologies will have on privacy and with which practices privacy can be enhanced.

27.2 The Reason behind the Prototype

The evolution of official credentials and other certificates to include electronic and computing technology became known with the general term e-government. The use of biometric certificates and other means of identity management in services and every day life is constantly increasing and applications reach many citizens. The transformation of personal information into electronic data raises new challenges as well as threats and risks particularly in large-scale applications. Recently there is a tendency in the airline and airport business to use such identification methods in order to allow frequent flyers with a trusted profile to access faster security controls. These methods can include latest technology including contactless smart cards with encrypted biometric information, and other credentials. These processes can also be privacy driven if there is control in the use and storage of personal data (including biometric data). Our prototype investigates the different aspects of such processes, which use identity devices for privacy-enhanced security controls.

This application trial served as an observatory both for new PIM challenges and for identifying opportunities and challenges in applying PRIME design principles and architecture components to device-centric applications that process personal data. A challenging characteristic of device-centric applications is the implicit nature of interactions between the user and applications by means of devices acting on their behalf. Also these devices (smart card, RFID) have low power and processing capabilities and hence we cannot expect that they can make complex policy decisions on what data to disclose in a particular context.

The architecture components as being developed in the PRIME Architecture (D14.2.a) and that potentially benefit from this application are:

- **Identity control component:** The function responsible for creating pseudonyms and for binding attributes to pseudonyms. In particular this concerns the configuration of the layered data structure for distinguishing different partial identities: the customer/financial layer (customer commercial data), the ID layer (passport data), the biometric layer (biometric template). These layers need to be defined in a PRIME console and subsequently transposed to the customer (smart) card. It also concerns the creation of the pseudonym attached to the RFID tag. In addition, this component performs the issuance of the trusted traveller credential.
- **Policy management component:** Policies that manage the transfer of data from the devices to the processing systems. During check-in, passport data should only be transferred if it concerns an APIS flight in which case an APIS manifest needs to be prepared. But more in general, all the distinct data layers (partial identities of the traveller) should only be used in a well-defined application context. This also concerns the data on the RFID tag (which probably will be limited to an identification number of the boarding card) and its linkability to a particular traveller. One of

the questions is whether a policy can be stored on a device. This seems unrealistic with current technology and therefore one option would be to perform the policy negotiation between end user and service on a console and subsequently transfer these from the console to the relevant servers. The policies then need to be enforced at the respective servers each time that a device interaction occurs.

• **Access control component:** Protects the data contained in the smart card and on the RFID tag. It evaluates the rules and decides on the access/transfer of the different data layers when the context complies with the policy rules. In addition, the trusted traveler credential will be used without disclosing the real identity of the passenger to the checking systems.

27.3   The Trusted Traveler Use Case Scenario

As described in the initial document of this application prototype, the scenario that models the process of a usual air traveler going to the airport and getting to the airplane has four stages. The enhanced scenario for the trusted traveler incorporates one more stage. The first stage is the registration process required to give the trusted traveler status to an airline client. Therefore the five stages for the trusted traveler departure are:

1. Enrollment;
2. Check in;
3. Passenger restricted area;
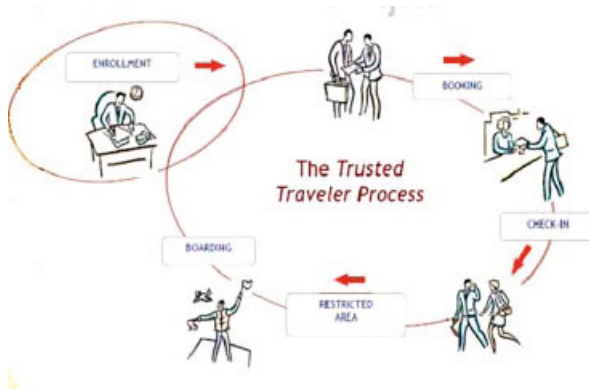4. Entering the gate; and
5. Boarding to the plane.

**Fig. 27.1** The "Trusted Traveler" scenario stages

27.3.1   Privacy Enhancements

During the modeling of the trusted traveler process, the following decisions were taken to enhance the privacy of the customer:

1. Biometric data are stored on the smart card (preferably only there) and are always encrypted.
2. A PIN can optionally be required to access biometric/passport data.
3. Biometric data are erased immediately after biometric verification.
4. Passport data are used only for compulsory APIS manifests.
5. Check-in personnel will not have access to personal data.
6. Further checks do not require presentation of any data to the staff.
7. The departure control may be automated and no further data from the customer is needed.
8. The RFID contains only a random number (transaction pseudonym of the passenger) and if authentication of the reader is used then the passenger cannot be tracked in an unauthorized way.
9. Duty free shopping can be done with only revealing the boarding pass RFID serial number.
10. Data storage, access and processing must be done in accordance with the privacy policy.
11. Data usage is logged.

27.4   Trusted Traveler "Smart Card" and Data Stored Therein

In the document [PRI04] describing the Application Scenario, the data required were presented. The data stored within the smart card, which is the
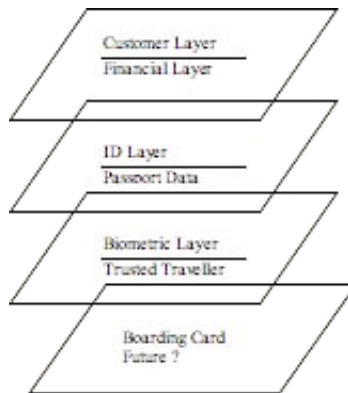
Fig. 27.2 The data layers for trusted traveler

basic token of the "trusted traveler", can be organized in three layers (groups) and in the future the boarding card layer could be added. The three layers are differentiated because they may require different handling (updating, integrity checking) and different levels of security. The data are collected with the customer's consent.

- **C**ustomer layer: Contains all the standard "frequent flyer" data as well as some data related to the credit card and account charging (e.g., customer ID, title, first name, surname, credit card data, and milage).
- **P**assport layer: Contains all the data that appear on the passport page. For an e-passport it can contain all the electronic data (after authority permission) which may include biometric data.
- **B**iometric layer: Stored only on the smartcard; can be any biometric method (e.g., based on fingerprint, iris, or face) depending on the method used in the scheme for authentication.
- **B**oarding card layer: Contains the data included in the common boarding card that eventually, if the official process allows for it, could be translated to electronic credentials. The data will be stored on the smartcard. Standard departure data will be stored in the airport's DCS. The validity is restricted to a time window sufficient for the departure (date, flight, 3-4 hours).

27.5 The ASC Prototype Stages

27.5.1 The Enrollment

The aim of the enrollment phase is to issue the trusted traveler card to selected airline passengers and to register the customers to the service. The process consists of the data capture and of the approval of the agreement between the passenger and the airline on the use of the smartcard and the data stored on it.

First the agreement is presented to the passenger. The agreement explains what data is captured at the enrollment, where the data is stored at what moment and how the data is processed during the departure control. The agreement includes also the privacy statement about the treatment of the personal data by the airline company. As soon as the passenger signs the agreement the process of data capture can start.

First the passport data is read from the passport (first name, surname, nationality, date and place of birth, gender, issued at/on and validity of the passport). The data can be obtained automatically using a reader of the machine readable zone or manually typed. In our prototype we will only use manual typing.

If the user wishes so, also long term visas can be stored in the passport layer of the trusted traveler card (airlines are responsible for checking visa necessity/validity at the time of check-in). In such a case the country code and the visa validity are saved. Another optional step is the frequent flyer card replacement. The trusted traveler smartcard can function also as a replacement of the classical frequent flyer card. In our prototype the frequent flyer functionality will not be implemented. Trusted traveler smartcard could also be integrated with a (co-branded) credit card (e.g., VISA). For obvious reasons such functionality will not be implemented in our prototype.

Biometric data capture is in fact a common biometric enrollment, where the user biometric characteristics are read, their quality is verified and then the raw biometric data is processed and the passenger's biometric template is created.

The data stored on the smartcard during the enrollment was organized in three layers: the customer (frequent flyer) layer, the passport layer and the biometric layer. Data in passport and biometric layer was digitally signed to preserve authenticity of the data and encrypted with a symmetric key to achieve confidentiality of the data. The RSA algorithm was used for asymmetric digital signing. The AES algorithm was used to symmetrically encrypt the data. To cover cryptographic functionality in our prototype we used opensource cryptographic library OpenSSL.

In addition to encryption the passport and biometric layer can be protected by a PIN. Such a PIN protection is optional and if activated it would not allow reading the passport and biometric layer without the passenger consent (entering the PIN), which will slow down the check-in, but increase the data protection. The decision whether to use PIN or not will be made at the time of enrollment and can be changed anytime later (PIN can be changed as well). The data in particular layers will be organized in records with fixed structure to maximally save (scarce) memory on the smartcard. The smartcard used in our prototype was dual interface (contact and contactless) Javacard EEPROM memory.

The airline maintains a central database of issued trusted traveler cards. Such a database does not, however, store personal data from the smartcard; it only associates a validity status with each trusted traveler card. At the time of check-in the validity of the card is verified and the card can be used only if it is valid. Reasons to revoke a card include above all situation when the card is lost or stolen. Revocation of the trusted traveler cards will be done from the enrollment office (hotline option). Issuance or revocation of cards requires on-line connection with the card database. In our prototype the enrollment point is connected to the card database and trusted traveler cards immediately are issued on the spot. In reality the smartcard might be issued within a few days and send by mail. Reasons might include the protection of the signing key (that will not be available at each enrollment office), background check of the passenger or enhanced printing on the smartcard surface. The trusted passenger can also decide not to use its status anymore (opt-out) by returning

his smartcard. The data in the central database and in the card are erased and the status of the card is changed to 'revoked'.

If some of the data stored on the smartcard need to be changed to reflect a change in reality then a visit of the enrollment office is necessary and based on the availability of the signing key the change either will be solved on the spot or within a few days.
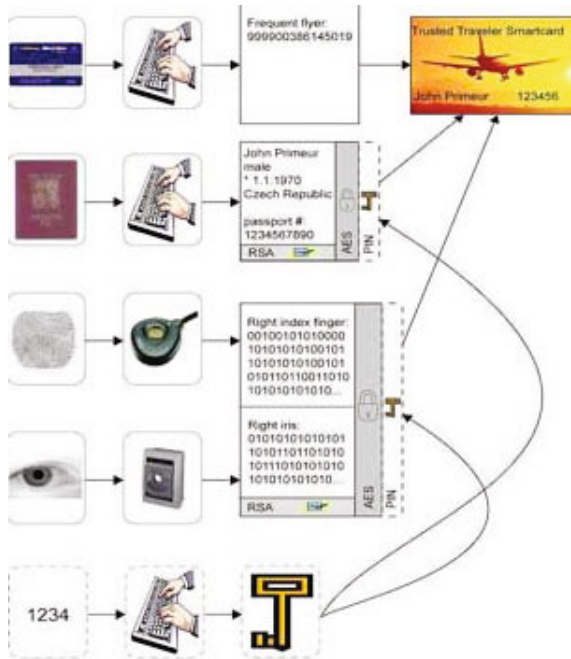

Fig. 27.3 Enrollment

27.5.2 Check-In

The purpose of the check-in is to issue boarding cards to passengers with valid airplane tickets.

First the passenger presents his airplane ticket: either traditional paper one or an electronic one, in the case of the electronic tickets either the ticket number or only an ID with the passenger name (or frequent flyer/credit card) is necessary. The trusted traveler scenario does not impose any changes on the booking process. He is asked to identify himself and if he has a "trusted traveler" card he can use the smartcard reader and the biometric reader to authenticate himself.

Next the boarding card is provided in traditional paper and the only difference is an RFID chip attached. The RFID tag is read-only and except for its serial number does not hold any other data. The presence of RFID tag will enable automated verification that the passenger has a valid boarding pass. The RFID tag number is only stored in the passenger's credentials on the smartcard so it is crosschecked at later stages. There is no central storage of the tags, so no passenger identification is possible without his trusted traveler card. Ideally the RFID tag number should be a random number and be revealed only to authenticated readers. Current technology has provided cryptographic RFID tags but this is far from being applied in large scales applications and therefore we used basic read-only 64bit RFID tags that will be attached to boarding cards.

The process of check-in begins 24 hours before the flight departs when the flight data (list of passengers) is transferred from the booking system (Computer Reservation System, CRS) to the check-in system (Departure Control System, DCS). Passengers being checked-in are searched in the DCS system to verify their tickets and seats allocated to them; when boarding cards are issued information in the DCS database is updated.

In our prototype we did not deal with the booking (CRS) system at all. The departure control system will be simulated to a certain extend necessary to make all the stages of the departure control work sufficiently well to be able to demonstrate the principles of the prototype. The departure control system was simulated in a form of a server to which other components are connected as clients. Although typical check-in counter offers through check-in to facilitate connecting flight, our prototype did not deal with transfers and other more complex situations like flight canceling.

Check-in counter is also the place where the luggage is handed over, its weight is checked and a luggage identifier is attached to it. Corresponding identifier can also be attached to the passenger's airplane ticket or boarding card and will be used if the luggage is misrouted. Luggage handling will not be simulated in our scenario.

For certain flights the Advanced Passenger Information System (APIS) collects information about flight passengers and when the aircraft door is closed, the list is forwarded to proper authorities (typically immigration control of the country of the destination). Because the data required for the APIS manifest are not all included in the PNR booking records, some data (like the birth date and nationality) must be captured at the time of check-in. This is normally done manually, but for trusted travelers the data capture can be automated using the smartcard.

The trusted traveler smartcard during check-in in more details: If the passenger has a valid ticket the smartcard is activated and the biometric layer is read. Reading of the biometric layer requires authentication of the smartcard reader and the data obtained from the smartcard must be decrypted and the signature of the biometric data layer must be verified. If the user decided to protect his biometric data with a PIN then the correct PIN must be entered
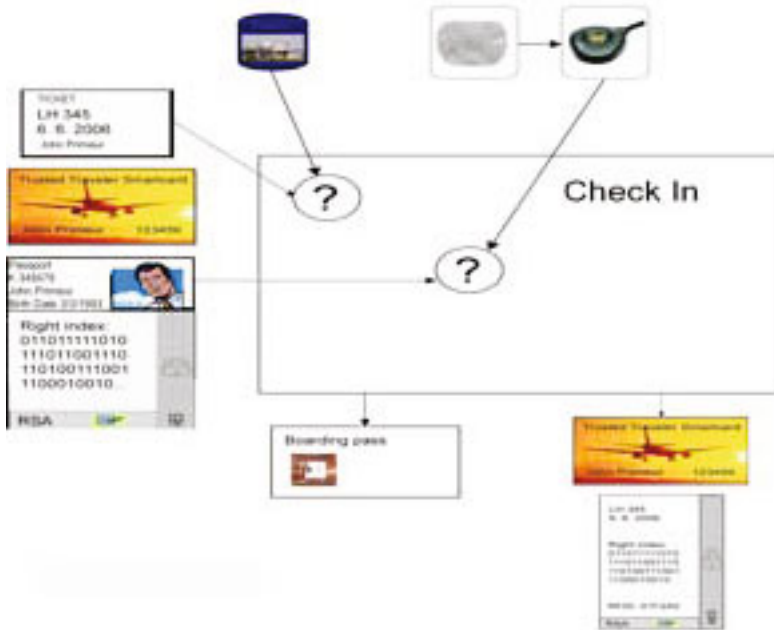
Fig. 27.4 Check-In

before any data is read from the smartcard. The PIN verification is done on the smartcard (in a way similar to SIM cards). If the data is successfully read from the smartcard and the traveler can verify himself, then the "virtual boarding card" in the form of credentials will be loaded onto the card. The credentials will include biometric data from the biometric layer, the tag number of the RFID chip, passenger ID, flight date and number. There are in fact several kinds of credentials for different purposes with various numbers of fields (see the data structures chapter 9 for more details). The credentials are digitally signed and symmetrically encrypted with a key that will be available to all airport devices that will need access to the credentials. Then the credentials are loaded onto the smartcard to the credentials layer.

27.5.3 Entering the Passenger Restricted Area (PRA)

Access to parts of the departure control area is permitted only to flying passengers. Entrance to the passenger restricted area is typically guarded by a person who manually examines boarding cards of entering passengers to check

whether they have been issued for a flight leaving from relevant PRA. Sometimes a photo ID document (like passport, driving license or ID card) is also required and then the names on the boarding card and on the ID are matched. Names may be checked against "black lists". Because of the manual nature of the check no log of entering/leaving passengers is maintained.

In the trusted traveler scenario the entrance to the PRA should ideally be combined with the check-in counter. The checked-in passenger could then directly enter the PRA without additional checks, which would speed up the departure control process. Naturally such arrangement of the check-in counter and the PRA entrance cannot be achieved at all airports. Therefore the trusted traveler scenario is using the passenger's smartcard and boarding pass to automatically verify the entry of a trusted traveler. If the airport structure does not allow for a separate entry at all (e.g., temporarily), then the legacy (common) entry must be used.

In the legacy PRA entrance there are two options: with or without ID check. Similarly in the trusted traveler scenario there are two options: with or without biometrics. If the legacy system performs only random ID check then the trusted traveler biometric verification could be also performed randomly.

To verify the passenger has got his boarding card with him we have to check for the RFID tag attached to it. Because the ID tag is not stored in any central database it is necessary to read the credentials of the passenger's smartcard to get the RFID number. If we need to biometrically verify the passenger then we have to read the "Biometric credentials" from the credentials layer of the smartcard, otherwise "Anonymous credentials" are sufficient.

After the credentials are read they are decrypted and the digital signature is verified. The credentials include the flight date and number, so verifying whether the passenger can be allowed in. Note that this is a scheduled date and it can change in reality (e.g., if the flight is delayed/postponed) therefore a database of flight to depart is necessary. Such a database could be offline uploaded from time to time or the gate can be online connected to the departure control system. In our prototype the PRA computer will be connected to the server where a database of flights will be located.

In this scenario we assume that the biometric verification is required at this stage, so the biometric template is extracted from the credentials and compared to biometric data acquired from the passenger on the spot. If the biometric data do not match the template, the user is notified and up to 4 other attempts are permitted. If the biometric verification does not succeed then a manual check of the ID must follow.

At the end the system checks whether the passenger carries his boarding card. The RFID chip number found in the credentials is being searched and then is matched with the passenger's boarding card RFID tag. The boarding card is not strictly necessary for departure control of trusted travelers but for compatibility reasons with the legacy systems it is being issued and its presence is checked even for trusted travelers.
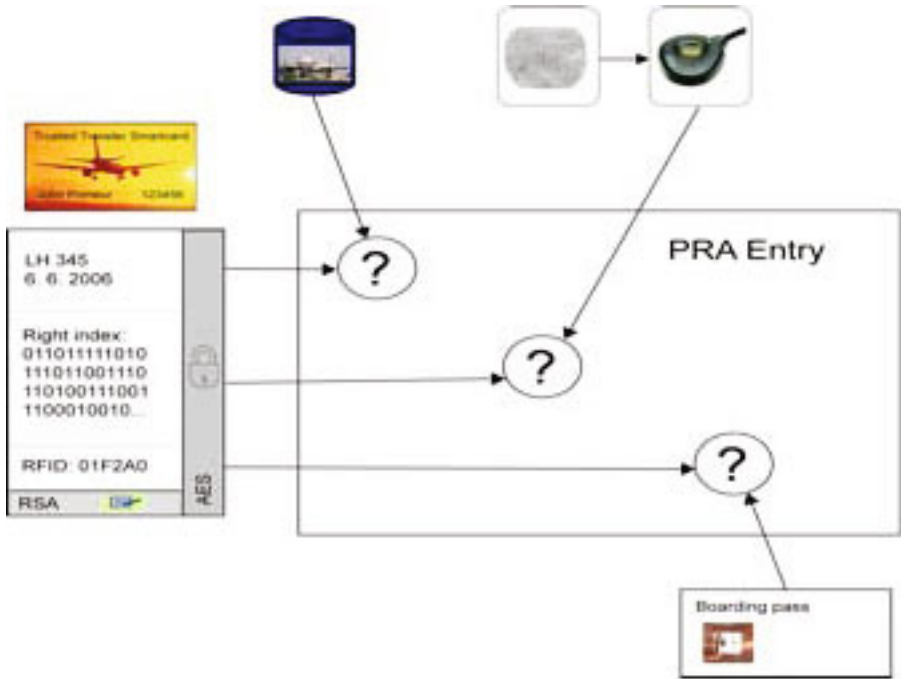
Fig. 27.5 Passenger Restricted Area

The entry of the passenger into the PRA is not logged and except for the above-mentioned database of flights-to-depart the check of the passenger can be done offline. In this prototype biometric authentication was required at the PRA entrance and the "biometric credentials" from the passenger's smartcard were used.

27.5.4 Gate

The security procedure at the gate varies from airport to airport. While the gate is open to enter without checks at some airports or the gate is actually not there at all (e.g., Milano Malpensa [MXP]), some airports do the personal security checks (i.e. X-ray of hand bags, body check) including the boarding card verification at the gate (e.g., Prague [PRG]). At some airports only the boarding card is checked and at some airports entering the gate is actually taken as boarding (e.g., Vienna [VIE]).

In the trusted traveler scenario the process of entering the gate will be similar to the normal passenger gate entrance. If no check is done at the gate, then also for trusted travelers there will no check at the gate. If security

screening is required at the gate then trusted travelers are also be screened (they will have their own channel – typically with a shorter queue).

In some cases only the boarding card is checked at the gate, so the trusted traveler scenario will use anonymous credentials to check whether the passenger is to depart from this particular gate. If in addition this, check of ID document is required, the name on the boarding pass has to match the name on the ID document. In this case the trusted traveler scenario will require biometric verification of the passenger. The gate use case can be considered a partial or identical case with the PRA and was not implemented in this prototype.

27.5.5   Boarding

Boarding is the final stage of the departure control. At the traditional boarding stage the passengers show the boarding card and an ID document (even for European flights within the Schengen zone). The names on the boarding card and the ID document are matched and the boarding card is split up. The boarding card coupon remains with the passenger and the other part of the boarding pass (including the flight coupon for non-electronic tickets) remains at the airport. The boarding passes of boarding passengers are fed into a machine to have a log of who is inside of the aircraft.

For trusted travellers the check of the ID document is replaced by an automated biometric verification. First the "identifying credentials" are loaded from the credentials layer of the passenger's smartcard. The "identifying credentials" are necessary at this point because the identity of the boarding passenger must be known. The credentials can only be read after successful authentication of the smartcard reader and the credentials obtained are decrypted and the digital signature is verified. The flight number and date in the credentials are compared with the actual flight and date the boarding is open for to make sure the passenger is boarding the right flight. Next the passenger is biometrically verified. The biometric template is found in the credentials and in-site biometric data from the passenger are matched against the biometric template.

At the end the passenger feeds the boarding card into a boarding card reader. The reader detaches the flight coupon and the boarding pass coupon is returned to the passenger. The part of the boarding pass with the RFID chip remains in the reader.

The data from the trusted traveler smartcard reader and the boarding pass reader are matched and a log entry is generated. The knowledge of who is inside of the aircraft and who checked-in and did not board at all is important to unload his baggage. Although more privacy-aware solutions for the baggage problem would be possible, there are other reasons and regulations why the identity of the passenger should be known. Unfortunately air travel cannot currently be as anonymous or pseudonymous as e.g., train or bus travel in EU is.
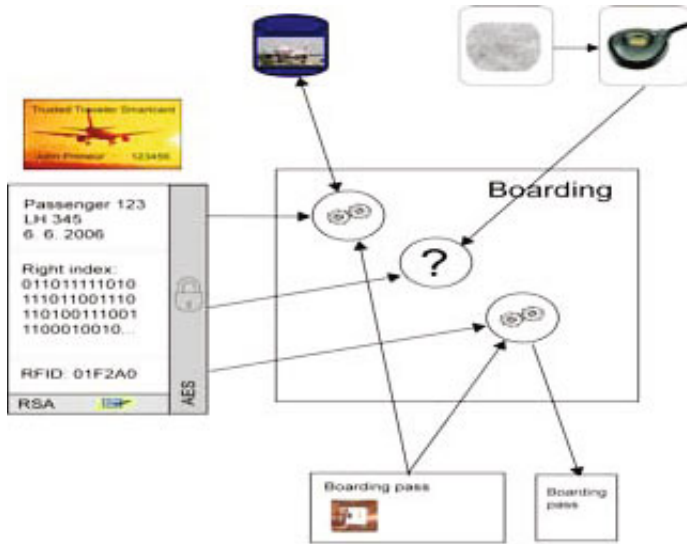
Fig. 27.6 Boarding

The boarding system needs to know flight date/number of the flight being boarded and the list of checked-in passengers. This list is updated with a flag, whether the passengers boarded or not. Although this can be done offline (e.g., a file on a diskette) the online connection to the airport DCS can facilitate the data exchange. In our prototype the boarding computer was on-line connected to the server where it could access the database of check-in passengers and update their boarding status.

When the aircraft door is closed the APIS manifest must be prepared for some flights. The manifest includes personal data captured at the time of check-in for passengers who boarded the flight. In our prototype we will generate a sample of such an APIS manifest in a text file for certain flights.

27.5.6   The Use of Cryptography

In our scenario cryptography is widely used to guarantee security. Digital signatures are used for integrity assurance. Symmetric encryption is used to achieve confidentiality of the date and symmetric keys are also used for smart-card reader authentication.

Digital signatures are used to verify integrity of data in biometric layer, passport layer and credentials layer. Data in biometric layer are signed by a private key S_B and the signature can be verified using the public key P_B. Data in passport layer are signed by the private key S_P and the signature

can be verified with the public key P_P. The private keys will be available only at the place where the trusted traveler smartcards will be issued (and its confidentiality will be guarded).

The anonymous credentials are signed by the private key S_{CA} and the signature can be verified with the public key P_{CA}. For confidentiality of the data symmetric encryption is used and access to some data records is allowed only after authentication of the smartcard reader.

Data in the biometric layer are encrypted with the secret symmetric key K_B. The encryption is done after the data are digitally signed.

Access to credentials is allowed only after the reader authentication and based on the A-keys in Table 27.1. Data signing and encryption as well as the data decryption and signature verification is done at computers, the smartcard functions as a data storage only. The authentication for the access control is done by a challenge-response authentication algorithm using symmetric encryption. The symmetric keys for authentication and encryption are diversified by a standard key diversification algorithm for particular smartcards.

The Table 27.1 summarizes which keys are used at which points. More detailed description for the key system can be found in [VR05].

Table 27.1 Cryptographic Keys needed. The key notation indicates where it is used and what type of key is: P = public, S = Private, K = Symmetric, and A = Access, with the subscript as follows B = Biometric, C = Credentials, P = Passport, W = write, R = Read, I = identifying, DEL = Delete, $DELALL$ = Delete All.

	Integrity	Confidentiality	Access control
Enrollment	S_B, S_P	K_B, K_P	A_{RB}, A_{RP} $A_{WCB}, A_{WCA}, A_{WCI}$ A_{DELALL}
Check-in	P_B, P_P S_{CB}, S_{CA}, S_{CI}	K_B, K_P K_{CA}, K_{CB}, K_{CI}	A_{RB}, A_{RP} $A_{WCB}, A_{WCA}, A_{WCI}$ $A_{RCB}, A_{RCA}, A_{RCI}$ A_{DEL}
PRA entry	P_{CB} or P_{CA}	K_{CB} or K_{CA}	A_{RCB} or A_{RCA}
Gate	P_{CB} or P_{CA}	K_{CB} or K_{CA}	A_{RCB} or A_{RCA}
Boarding	P_{CI}	K_{CI}	A_{RCI} A_{DEL}

Privacy and Identity Management Requirements: An Application Prototype Perspective

Tobias Kölsch[1], Jan Zibuschka[2], and Kai Rannenberg[3]

[1] T-Mobile
[2] Fraunhofer IAO
[3] JWG Universität Frankfurt

28.1   Introduction

The following requirements, which have been derived from the previously described application prototype scenarios, constitute features that a privacy-friendly identity management system should support, based on a wide range of application scenarios. However, the requirements should not be read as absolute: some of them might be impractical or even impossible to implement in a specific context, e.g. because of high system resource requirements with regard to their operation or difficulty of implementation. Rather, this chapter contributes experiences from the application prototype development to the ongoing discussion of generic identity management requirements. It also shows to a certain extent how PRIME could meet those requirements. Still, it always tries to stay generic in its descriptions, to ease comparison between scenarios and with other scenarios. The prototypes presented in the preceding chapters include examples of complex services that are offered by consortia; so, for example, more than just the two "classic" stakeholders (customer, provider) may be involved in, e.g., location-based services. For a more detailed discussion of particular aspects of this, refer to the chapters on the individual prototypes of Part IV and to selected chapters in Part III of the book, e.g. Chapter 22 for multilateral user-to-user interactions. This chapter will briefly list the main requirements for the key stakeholders identified in the scenarios and then elaborate on them. The stakeholders presented here are:

- User
- Service Provider

- Network Operator
- Developer

This is based on observations during the requirements gathering process, leading to a categorization of stakeholders and a multilateral analysis of their requirements (for a more thorough discussion of the LBS case, see [ZFR$^+$07a]). Those stakeholders and the identified requirements seem also to hold relatively well in other information and communication technology-related scenarios not directly covered here, e.g. the web.

28.2 Users' Interests and Requirements

Users in many cases are the weakest of all the parties mentioned, especially if each user is acting as an individual. So, their requirements concentrate on the ability to retain control of their data. They are structured within the categories:

- Data Minimization
- Control of Data Flow
- Easy-to-Use Technology
- Reliable Service Provision

28.2.1   Data Minimization

28.2.1.1   Confidentiality of Service Utilization

A network operator often is in a position where all the service usage patterns of its customers could, at least in principle, be monitored. A similar case is made by the hospital scenario, using classical mail. PRIME's Identity Management architecture offers a flexible solution to this problem (see "Pseudonyms" below).

28.2.1.2   Anonymity

Users do not want to reveal their identity unnecessarily and, in many cases, would like to stay anonymous. The PRIME system is tailored towards this requirement and could be integrated into prototypes in a wide variety of scenarios, even offering several implementation options, which were necessary, e.g. for the WAP LBS implementation.

28.2.1.3   Protection of Service Configuration Data

Users don't want other parties to unnecessarily know what interests they have, e.g. what services they use. Specifically, the question of service parameters, that may, e.g. be medical information, and their handling turned up several times in the PRIME application prototypes.

28.2.1.4   Pseudonyms

For distributed services it is often important to have a way to refer to a user by a pseudonym, as the true identity of the user should be protected. Some kind of pseudonyms have to be offered to create a relation with the user that goes beyond a one-shot transaction (in this case the network connection can be seen as an implicit pseudonym). Obviously, these pseudonyms should be unique and unforgeable and serve as a well-defined reference to one or several service relations between the service provider(s) and the user. The unforgeability should be provided by cryptographic means, as this is the only reliable means to prevent man-in-the-middle attacks against the user. However, some alias mechanism should also be available to permit using mnemonics for pseudonyms (e.g. Gummibärchen@jabber.org) that are easier to handle for the user than the internal pseudonyms. This is especially important for community services, for example a friend finder application. Of course this matching can also be performed on the application layer. The pseudonyms should also be usable as reference between third parties, so that different external instances have a means to refer to the user without being able to impersonate him. Pseudonyms are closely linked to partial identities. Partial identities represent everything that is known of some individual within a service. Pseudonyms provide a way to refer to some partial identity that has been established during an earlier service provision. Obviously, there is also a strong link between partial identities and service contexts, as all information that is released to some partner within a context could be assembled to a partial identity. Sometimes it is also convenient to link different partial identities together, e.g. to combine partial identities from different services a user is registered for.

28.2.1.5   Credentials

Identity management systems deal with transmission of personal information. This personal information can be seen as credentials. A privacy-enhanced identity management system should support credentials with different assertion levels. The simplest form of credential is similar to an HTML form in web-applications, it is just a simple unverifiable claim made by the user. However, in critical applications that depend on the claims made by the user, it is important to have some higher level of assertion on the correctness of the transmitted information. Cryptographic credentials can be used to serve this goal. There are quite different types of credential systems. One possibility are classical Circle of Trust-based federated identity management systems. Another class is given by assertions based on asymmetric cryptographic signature schemes (PKI). The third possibility is using advanced anonymous credential systems with untraceability and features such as n-show restrictions (the credential may be shown at most n times). These systems provide

different modes of operation that have different costs with respect to implementation, setup and operation effort. Generally, a credential system for PIM should support different modes due to varying security, performance and interoperability requirements in different application scenarios. The plain user-provided information modes should be supported along with at least two of the following: signature-based certification by identified user, signature based certification by third party, certified anonymous credentials (optionally with the possibility of anonymity revocation for dispute resolution). This diversity is especially important, as the modes are quite different with respect to the setup costs and with respect to their performance. To illustrate the need for different modes, regard the following example: In the pollen use case also employed for the LBS application prototype (see Chapter 25) it is not really important to assert that the allergy profile provided by the user is correct, in fact it might even be serviceable to give the user the possibility to set some profile that is not his one. If we wanted to inhibit this, he would have to have some trusted physicist assert his allergy profile before using the service. On the other hand, for the same service, if it comes to connecting to the MO to update the access policies, it is important to assert that the connecting entity is really the one it claims to be. This guarantee can be provided by credentials. As with most cryptographic systems, the setup and maintenance of a credential system can be difficult for all involved parties. The IDM system should minimize the burden for all of them, but especially the overhead for the user should be minimized, as he will usually not be an expert in identity management. Catchy metaphors can help here. The management operations that must be supported are: support for issuing, expiry, and revocation of credentials. Additionally, there should be support for offline credentials that do not require a network connection to some trusted third party. In case of a dispute or fraud, secure dispute resolution mechanisms (e.g. to revoke the user anonymity) should be provided.

28.2.1.6   Protected Communication

As all data is transmitted over communication networks, it is obvious that the communication has to be protected. This protection must happen at different levels. First, the traffic data should be hidden from eavesdroppers, such that little or no information can be gained from observing a user's internet connection. For this, some anonymization system has to be supported. The overhead that results from the anonymization should be adaptable to the security requirements of the application. Ideally, it should not be visible to an outsider whether a highly-secure connection or a less secure connection has been chosen. Second, the traffic content has to be protected. This can be best done using some established content encryption mechanism. This functionality is usually already present in libraries nowadays. Third, the traffic content must be reliably associable with the respective sender. This can be done by associating the stream cryptographically to the identity information

that is exchanged at the application level (e.g. using the pseudonyms in X.509 certificates for the connection handshaking).

28.2.2   Control of Data Flow

28.2.2.1   Sovereignty over Personal Information

Users require facilities to configure the acceptable usage of their personal information. Even in cases where the transfer of personal information is legitimate, the user often wants to stay in control of the transaction and is also supported in this by regulation (see Chapter 5).

28.2.2.2   Fine-Granular Management of Consent

The LBS scenario is an example of dynamic personal information in a complex multi-party scenario requiring a relatively fine-grained control system. Users may want to configure specific parameters concerning the handling of their location information by different LBS providers or may only want to be monitored during certain times of day. Thus, it is an example of a scenario where users desire a relatively fine-grained consent policy management.

28.2.2.3   Context Awareness

For privacy-friendly services it is imperative that each service relation is aware of the actual context, as privacy is essentially context bound (i.e., it is OK to communicate business secrets within the company, but not outside; the health record should be made available to a treating physician, but not to the employer). So it is important that a generic privacy architecture such as the one offered by PRIME supports context awareness in a way that prevents information from leaking from one application to the other. Additionally, the management of services in terms of service contexts is much more intuitive to a user, as this way, using the transaction logs, a user can monitor which information has been transmitted to a provider in the context of a specific service. The context also helps the user to keep track of the providers involved in the provision of this service.

28.2.2.4   Privacy Policies

One important subject in identity management and on-line service usage is access control. The user wants to control the information that is disclosed to the service providers, whereas the service providers often require that a service is only made available to a user if he fulfills some criteria. As the specific usage of a generic PIM architecture cannot be fully foreseen by the developers, the policy system must permit for maximum flexibility within

reasonable boundaries, set by performance. It must be able to restrict access based on attributes (credentials) of the peer. The values of the credentials (e.g. the asserted age of a user) must be comparable to threshold values or to context variables, for example denying access after 22:00 h. Furthermore, the context functions should be easily extendable by the application developers. It should be possible to restrict the access to personal information on different granularities. This can be done in combination with ontologies. So the access control could be done for each data item or for groups of items (e.g. "grant MobileDating access to my hobbies"...). Sometimes it might be sufficient to "blur" the data that is disclosed. This is similar to the telephone bill from the Deutsche Telekom that can optionally have the last 4 digits replaced by 'x'. The policies could specify some precision restrictions on the disclosed data. For example, it could be specified that the location is only disclosed with an uncertainty of 50 km. If ontologies are supported, it could be specified that some specific data types are not disclosed directly. Instead, only relations are returned (e.g., textrmage < 65). The blurring filters that can be set should be extendable by the application developer. From the services side it should be possible to set basic policies that are applied on each access to warrant for base restrictions, for example to comply with regulation. So, all accesses to a server for x-rated content could be restricted to users that prove they are of age. Policies will often be application specific, which strongly restricts the possible values of the policies within a specific application. This restriction is important to give the system operators and the users more guidance in how to configure their policies. On the user side, this can be done using a policy definition template that restricts the possible policies and specifies value ranges for the different attributes. These application-specific restrictions make it possible for the application developer to design application-specific policy editors that present customized editing views to the user.

28.2.2.5   Data Lifecycle Management

In a PIM system, it makes sense that the user has the possibility to control access to his data beyond the scope of his device. So he may disclose some information to a service provider, but specify some storage time constraints. He might require logging of access to his data, or he might require notification on access. All these functions should be supported. Policies to specify the desired behavior have to be provided. In PRIME nomenclature these policies are called obligations. However, the user is not the only one who might want to specify restrictions. It could also be necessary to implement restrictions on the services side, for example to clean up the own database some time after the service usage, or to defer the deletion of data beyond some legal retention period.

28.2.3 Easy-to-Use Technology

Privacy functions should not impede usability, especially not the usability of mobile services, as those services are usually used in settings where users cannot simply concentrate on dealing with the user interfaces. Ease of use is an important requirement both for adoption and for the effective security offered by the system when operated by a human.

28.2.3.1 User Interfaces

Different user categories that are created by the various contexts and backgrounds of users, such as PC user, mobile phone user, knowledgable user, average user and so on, create a necessity for good metaphors for the different privacy operations, as well as pervasive availability of the system. The identity management must not lay a heavy burden on the end user, as this would severely hinder the market diffusion of the technology. So, the metaphors must employ widely-understandable concepts. But the developers and the administrators must also feel comfortable with the employed technologies and mechanisms, so that development costs are not excessively increased. This is especially true for generally difficult-to-grasp concepts such as policies, ontologies, and (anonymous) credentials. PRIME's approach to those questions is described in more detail in Chapter 20.

28.2.3.2 Data Management Transparency

As the user is the "owner" of his personal data, he should be enabled to view what processing is performed on his PII. This is a simple means to increase the user's trust. For this, the PIM system should log all relevant accesses to the user data and provide means for him to scrutinize these access logs later on. Deciding which accesses are relevant and finding a balance between transparency on the one hand and performance, usability, and not disclosing business secrets on the other hand is a difficult problem. Logging all accesses would offer the best transparency. However, it can constitute a quite important performance issue on highly-frequented systems. Also, applications that continue processing in the background would produce a huge amount of log entries for one user (a tracking service that locates the user every 15 minutes would produce 672 log entries a week) that could hide relevant entries from the user. So, which accesses are "relevant" is not necessarily obvious to the PIM system. As a result, it should provide means for defining which accesses should be logged and which ones should not. Obviously, this feature might be misused, but not permitting configuration here might render the entire system unusable for certain applications. In addition to defining which automatically-generated entries are relevant, the application should also be able to add custom logging entries. By this, conglomerate accesses could be combined to a single logging entry instead. For example, in a mobile scenario,

a user positioning request might for example consist of a number of secondary requests:

1. First the application has to retrieve the MSISDN of a user that is referred to by his pseudonym;
2. then it must be identified which mobile operator the user has subscribed to;
3. some access credentials for requesting the user location from the MO might be retrieved;
4. then the user location would be retrieved;
5. and finally forwarded to the requester.

From the point of view of the user, it does not make much sense to log all 5 entries for each location request; it would increase the clarity to replace them by one application-specific entry.

28.2.3.3   Remote Administration of User PII and Policies

In the course of a service, the user submits data to the service provider. However, this information may vary over time, such that an update might be necessary. Additionally, the policies that have been submitted might not be adequate anymore after a while. To reach this goal, the PIM should provide means to administer the policies and the user information. One possibility would be to support this directly in the PIM, but it is also thinkable to have an interface to permit for application-level remote user data and policy management.

28.2.4   Reliable Service Provision

Availability of the service is a major concern, especially in scenarios such as LBS search and navigate, where the PRIME system is used in the context of a service employed by the user to save time.

28.3   Service Providers' Interests and Requirements

The service provider's requirements focus on running his business easily and securely. For services, it is central to meet user requirements to build a base of trusting users. Furthermore, they want access to identity management interfaces at operators' walled gardens to access a substantial user base in a well-defined and compliant way. Services' main interests as presented here are:

- Flexible Business Models
- Customer Loyalty and Trust
- User Base

- Trusted Payment Partners
- Legal Compliance
- Delegation

28.3.1   Flexible Business Models

As different telecommunications markets favour different organizational structures, an architecture supporting several deployment structures is essential for real-life deployment of the architecture. This is enabled, e.g., by the modular architecture of the prototype.

28.3.2   Customer Loyalty and Trust

Both service providers and network operators value customer loyalty, which may be increased by respecting each customer's privacy, which may be realized by implementing privacy-enhancing technologies at the service provider. So, the service provider may realize an improvement in this area by supporting users' privacy preferences using PRIME.

28.3.3   User Base

It is imperative for the service provider to have easy access to the largest possible user base (e.g. often via the mobile operator in mobile scenarios).

28.3.3.1   Internationalization

One way to increase the user base is by providing multi-lingual support. The PIM should also support internationalization, at least for all parts that the user gets in contact with. This includes the policies, the ontologies, and the access logs.

28.3.3.2   Bridge to Legacy Applications

In the context of LBS, applications have a rich environment of existing infrastructure. First of all, there is the existing GSM/UMTS infrastructure that is used to query the user location, on the other hand, many service providers already have some interface to third-party application providers. These interfaces can provide payment functionality etc. Some of these interfaces may (more or less easily) be switched to PIM-technologies. Others, however, may not. The location retrieval from the mobile network must, for obvious reasons, remain in the mobile network. A truly general-purpose IdM system must account for this by offering interfaces for redirecting data access as a delegated operation, so that it is able to manage personal information controlled by third-party identity providers, as is the case in LBS location retrieval. By this, the architecture can be built into such use cases and provide a unified interface for all applications.

28.3.3.3   Standardized Communication Interfaces

A standardized interface for management of identities, e.g. mediation of location information, is a requirement for tapping the network effect immanent in distributed multi-party scenarios. There are various benefits in this, e.g. for the case of LBS: mobile operator independence, roaming support, and the unified interface for service providers for easy deployment and migration seem to be viable business propositions in a fast-moving marketplace. Mobility between different services, involved market players, and applications seems beneficial from users' and service providers' perspective alike. From an ordinary user's point of view, cost effectiveness, synergy effects, and convenient service usage are major issues. Additionally, location sources are not limited to mobile operators. Depending on use cases and available technology, location information may be aggregated from several sources employing technologies like GPS, Galileo, COO, WLAN, or from several mobile operators. This improves the accuracy of delivered location information, and might even become a requirement in a world of converging network technologies. However, involving an independent location intermediary may be seen as undermining privacy, requiring special care.

28.3.3.4   Ontologies

The policies described in the previous section make statements on the handling of data types. These data types have no semantics by themselves. By introducing ontologies that establish a relation between the different data types, it is possible to significantly reduce the amount of policies needed (and by this reduce the configuration burden on users and system administrators). An ontology makes it possible to express "is a"-generalizations that can later on be used in policies to define restrictions on classes of data types. Support for this could be provided by a PIM. These ontologies should be extendable, as the PIM system designer cannot know all dependencies that might occur in an application. It should also be possible to turn it off if they do not make sense in the application at hand. A more advanced feature is that a request for a credential of an abstract data type can be fulfilled by presenting a credential certifying any specific data type that is in an "is a"-relation with that abstract data type. The classic example for this is proving the possession of a driver's license when a proof of age is requested (this at least works in some countries).

28.3.4   Trusted Payment Partners

As PIM is closely related to on-line service provision, the identity management system should also provide means for secure payment. Of course, for this, there has to be a high level of security and reliability. The exchange

should (as for credentials) support several modes of operation. Some low-value micro payments should be supported as well as highly-secure payment for higher-value transactions. The payment system should provide support for anonymous exchange, protecting the relationship between the payer and the payee against third parties, and optionally even for hiding the identity of the respective partner from the two involved parties themselves.

28.3.5 Delegation

In modern systems and service oriented architectures, it is common that an application is spread over different hosts or even companies. However, this means delegation of access control decisions, as some personal data is not stored and protected by the users themselves, but by delegate identity providers. This has to be covered by the policies and by the obligations. The expressiveness of the policies must suffice to specify disclosure behavior of a service provider's customers' data to third parties. For the obligations, there must exist a well-defined behavior specification on how the data is handled by the third party. In a B2B scenario, where one service agglomerates a number of other services, it is important to support the extension points for data access, such that requests for data can be specified as coming from another service provider, or as coming from some internal function.

28.3.6 Legal Compliance

Both service providers and network operators require that the interfaces provided by operators or identity providers are compliant with (potentially divergent) privacy legislation.

28.4 Network Operators' Interests and Requirements

In many use cases, such as LBS or the Web, a network operator comes into play in addition to the service/user pair. The main interests from the point of view of the network operator are focussed on enabling business models and increasing customer loyalty, and thus are conceptually very similar to a service provider's requirements, although on a different level. The main interests of a network operator we identified are:

- Flexible Business Models
- Easy Integration of Third-Party Services
- Legal Compliance
- Customer Loyalty and Trust
- Leveraging Existing Infrastructure
- Enabling New Applications

28.4.1   Flexible Business Models

As different telecommunications markets favour different organizational struc-
tures, an architecture supporting several deployment structures is essential for
real-life deployment of the architecture. This is enabled, for example, by the
modular architecture of the prototype.

28.4.2   Easy Integration of Third-Party Services

Easy integration of third-party services is a requirement for the service op-
erators, but also for the mobile operator. By offering an attractive service
portfolio to the user in cooperation with third-party developers, he can gen-
erate business while outsourcing risks.

28.4.2.1   Open Protocols

As it can never be fully foreseen which platforms may want to implement inter-
faces to the PIM system later, the communication protocols used to exchange
PIM and application data should be well-defined and open, such that devel-
opers from special areas can implement them (or part of them) to perform
operations that are relevant to identity management. A standardized inter-
face available at the different location sources provides flexibility and limits
deployment costs. Standardized interfaces can enable the network operator to
offer a wide range of externally-rendered location-based services to its users.
At the same time, it may lower costs for services.

28.4.2.2   Performance

In academia performance is not always an issue. However, when designing a
system for productive operation, it is important that the execution time and
memory consumption for operation is kept at reasonable levels. Additionally,
the worst- case behavior should be subject to some reasonable bounds. It is
clear that this can be complicated to accomplish in presence of ontologies,
as evaluations in ontologies can quickly run out of control if no special care
is taken in the design. This is a problem that should be quite relevant to
the application developers. In LBSs, the user device is usually some low-
performance mobile phone with limited input, output, and communication
capabilities. It is utopian to assume that the client that a PIM system is
meant for is exclusively the user's personal computer. The system could also
be installed on mobile phones, exclusively or in addition to a desktop variant.
So the PIM system should be sufficiently flexible not to require installation of
all features. A mobile phone implementation could go without ontologies and
with restricted policy editing and credential capabilities, for example.

28.4.3   Legal Compliance

Both service providers and network operator require that the interface he provides is compliant with (potentially divergent) privacy legislation.

28.4.4   Customer Loyalty and Trust

Both services and network operators value customer loyalty, which may be increased by respecting each customer's privacy, and thus building trust in the service provider.

28.4.5   Leveraging Existing Infrastructural Assets

Network operators are already managing their users' identities to a certain extent, holding personal data of non-anonymous subscribers, location information, device information, etc. Also, the billing infrastructure available at the network operator may be leveraged for payment services.

28.4.6   Enabling New Applications

Identity management deployments may serve as an enabler for new products, that could not be deployed without them, or at least would face problems with user acceptance.

28.5   Developer Requirements

To support effective usage of the technology, it is imperative to have a system that is understandable to the software architects and other developers. If this is not the case, the protection of the user cannot be warranted for. Also, it is often in the interest of other players in the field to attract third-party developers whose applications will improve the usefulness of their products by acting as a complement. The main requirements voiced by developers during the application prototype implementation were:

- Documentation
- Lean Interfaces
- Integration into Existing Frameworks

28.5.1   Documentation

Such a technology has to be described at different levels. First some high-level overview documentation is needed to effectively work with the complex conglomerate of technologies given by a PIM system. Such a description can also be used by the decision makers to base their technology choice on. The

architects and developers then need a concise and technical description of the different modules and their usage, to be able to design the application. This document should also contain a description of all interfaces, along with functions and their parameters. By providing some functional examples for the most common cases, the developers get concise guidelines on how to use the technology. And the learning curve can be flattened significantly.

28.5.2   Lean Interfaces

Even though the protection technologies can be quite complex, the application interfaces should be as concise and simple as possible. Normal setup procedures and general data access operations should be hidden behind commodity interfaces. Such that the complex low-level interfaces only have to be touched for advanced use cases. The interfaces should, if possible, reflect the metaphors that have been created for the operations. Of course, the requirements presented in this chapter are not that important in a research project, as it has the goal to explore new technologies. But they should be taken into account when going from a research prototype to deploying a system in a production environment.

28.5.3   Integration into Existing Frameworks

As a means of improving software productivity, many software projects are nowadays developed within some application frameworks or application servers, such as the servers specified by the Java Servlet API [4]. This should be accounted for by the PIM system, especially if it takes the form of a middleware. This means that it should provide interfaces for session management and for the access control and security mechanisms that can be integrated into application frameworks. The interfaces for data access should be compatible with common standards. Also, the PIM system should make no assumptions regarding concurrency and it should provide thread safeness on all functions (or thread safe alternatives for non-safe functions). It is clear that integrating the PIM into some application framework constitutes a major effort and cannot be required from the PIM system (especially, as there is a large amount of application frameworks). However, the design should keep an optional integration into application frameworks in mind.

28.6   Conclusion

As should be obvious from the content of this book, identity management systems have quite a wide application field. Specifically, we have found during the development of the PRIME application prototypes that it can empower the stakeholders in the investigated scenarios:

- Users can exert better control over (identity) data flows,
- Service Providers can offer personalized services to a large user base,
- Network Operators can act as the users' identity intermediary, and
- Third Party Developers can offer applications taking advantage of strong privacy-enhancing identity management

The PRIME architecture offers a set of technologies, presented in the earlier sections of the book, that are both comprehensive and state-of-the-art. The application prototypes demonstrated that those technologies can be implemented in viable application settings, to the benefit of all involved parties.

This generic analysis, together with the prototype scenarios presented earlier, demonstrates, how the PRIME architecture was used to meet privacy-related requirements from a wide range of disciplines and helped develop both novel applications and prepare for real-world deployments. While the developers had some additional requirements addressing the maturity of the integrated prototype, such as performance and documentation issues, those obstacles in the end did not hinder them from implementing strong privacy-preserving application solutions based on PRIME components. This validates the concept of a generalized identity management architecture that is not dependent on a single protocol or implementation, but is flexible enough to support a wide range of use cases, technologies, and stakeholder configurations.

References

[AFR05]   Albers, A., Figge, S., Radmacher, M.: Loc3 architecture proposal for efficient subscriber localisation in mobile commerce infrastructures. In: The Second IEEE International Workshop on Mobile Commerce and Services, WMCS 2005, Munich, July 19, pp. 154–162. IEEE, Los Alamitos (2005)

[Ahm03]   Rizwanuddin Ahmad, S.: Adverse Drug Event Monitoring at the Food and Drug Administration - Your report can make a difference. J. Gen. Intern. Med. 1(18), 57–60 (2003)

[BDF⁺05a]   Borcea, K., Donker, H., Franz, E., Liesebach, K., Pfitzmann, A., Wahrig, H.: Intra-Application Partitioning of Personal Data. In: Workshop on Privacy-Enhanced Personalization (PEP 2005), Edinburgh, UK (2005)

[BDF⁺05b]   Borcea, K., Donker, H., Franz, E., Pfitzmann, A., Wahrig, H.: Towards privacy-aware eLearning. In: Danezis, G., Martin, D. (eds.) PET 2005. LNCS, vol. 3856, pp. 167–178. Springer, Heidelberg (2006)

[BFD⁺05]   Borcea, K., Franz, E., Donker, H., Pfitzmann, A., Wahrig, H.: Privacy-Aware eLearning: Why and How. In: Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005, Montreal, Canada, pp. 1466–1472 (June 2005)

[BPFP05]   Borcea-Pfitzmann, K., Franz, E., Pfitzmann, A.: Usable Presentation of Secure Pseudonyms. In: DIM 2005: Proceedings of the 2005 Workshop on Digital Identity Management, Fairfax, VA, USA, pp. 70–76. ACM Press, New York (2005)

[BPL07]   Borcea-Pfitzmann, K., Liesebach, K.: The concept of workspaces – re-defined for eLearning. Advanced Technology for Learning (ATL) International Journal, Special Issue on Technology for Collaborative Learning 4(1) (2007)

[BPLW05]   Borcea-Pfitzmann, K., Liesebach, K., Wahrig, H.: Das Workspace-Konzept – neu verpackt im eLearning! In: 2nd Workshop "Structured eLearning", Pre-Conference together with DeLFI 2005 and GMW 2005, Workshop Proceedings DeLFI 2005 und GMW 2005, Rostock (September 2005)

[BPS07]   Borcea-Pfitzmann, K., Stange, A.-K.: Privacy – an Issue for eLearn-ing?: A Trend Analysis Reflecting the Attitude of European eLearning Users. Technical Report TUD-FI07-01, Technische Universität Dres-den, Institut für Systemarchitektur (February 2007)

[Che73]   Chernoff, H.: The Use of Faces to Represent Points in k-Dimensional Space Graphically. Journal of American Statistical Association (68), 361–368 (1973)

[CL01]   Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revoca-tion. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)

[CPHH02]   Clauß, S., Pfitzmann, A., Hansen, M., Van Herreweghen, E.: Privacy-Enhancing Identity Management. The IPTS Report 67, pp. 8–16 (2002)

[CRA99]   Cranor, L.F., Reagle, J., Ackerman, M.S.: Beyond Concern: Under-standing Net Users Attitudes About Online Privacy. AT&T Labs-Research Technical Report TR 99.4.3 (April 1999)

[DG07]   DiMasi, Grabowski: The cost of Biopharmaceutical R&D: is Biotech different? Managerial and Decision Economics (28), 469–479 (2007)

[dLvH08]   de Langen, J., van Hunsel, F.: Adverse Drug Reaction Reporting by Patients in the Netherlands: Three Years of Experience. Drug Safety 31(6), 515–524 (2008)

[EFP08]   EFPIA. The pharmaceutical industry in figures, 2008 edn. Technical report, European Federation of Pharmaceutical Industries and Asso-ciations (2008)

[Eur95]   European Parliament. Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individ-uals with regard to the processing of personal data and on the free movement of such data (1995)

[Eur02]   European Parliament. Directive 2002/58/ec of the european parlia-ment and of the council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic commu-nications sector (directive on privacy and electronic communications) (2002)

[Exp03]   Expert Consultation. Diet, nutrition and the prevention of chronic diseases. Technical report, WHO/FAO (2003)

[FBP06]   Franz, E., Borcea-Pfitzmann, K.: Intra-Application Partitioning in an eLearning Environment – A Discussion of Critical Aspects. In: Proceedings of the The First International Conference on Availability, Reliability and Security, ARES 2006, pp. 872–878. Vienna University of Technology, Austria (2006)

[FE06]   Franz, E., Engel, B.: A Realization of Context Management Facili-tating the Usage of Partial Identities. In: Proceedings of PEP 2006, Montreal, Canada (2006)

[Fig04]   Figge, S.: Situation-dependent services - a challenge for mobile net-work operators. Journal of Business Research 57, 1416–1422 (2004)

[FLBP06]   Franz, E., Liesebach, K., Borcea-Pfitzmann, K.: Privacy-Aware User Interfaces within Collaborative Environments. In: CAI 2006: Proceed-ings of the International Workshop in Conjunction with AVI 2006 on Context in Advanced Interfaces, pp. 45–48 (2006)

[FWBBP06]   Franz, E., Wahrig, H., Boettcher, A., Borcea-Pfitzmann, K.: Access Control in a Privacy-Aware eLearning Environment. In: Proceedings of the The First International Conference on Availability, Reliability and Security, ARES 2006, pp. 879–886. Vienna University of Technology, Austria (2006)

[Goo08]   Google. Google health beta website (2008),
            https://www.google.com/health/

[Hea05]   Health Canada. Buying drugs over the internet — it's your health. Technical report, Health Canada (2005)

[HR89]   Manasse, H.R.: Medication use in an imperfect world: drug misadventuring as an issue of public policy, part 1. Am. J. Hosp. Pharm. (46), 929–944 (1989)

[INR97]   INRA. Eurobarometer 46.1: Information Technology and Data Privacy. Report produced for the European Commission, Directorate General "Internal market and financial services" by INRA (EUROPE) - E.C.O. (January 1997)

[Int05]   International Obesity Task Force. Eu platform briefing paper. Technical report, International Obesity Task Force (March 2005)

[KBG04]   Kettel, L., Brooks, C.A., Greer, J.E.: Supporting Privacy in E-Learning with Semantic Streams. In: Proceedings of Second Annual Conference on Privacy, Security and Trust, Wu Centre, University of New Brunswick, Fredericton, New Brunswick, Canada, October 13-15, pp. 59–67 (2004)

[Kel08]   Kelly, W.N.: How Can I Recognize an Adverse Drug Event? Medscape Pharmacists (December 2008),
            http://www.medscape.com/pharmacists

[KFF⁺07]   Nagaraja, S.: Anonymity in the wild: Mixes on unstructured networks. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 254–271. Springer, Heidelberg (2007)

[KFKK05]   Kölsch, M., Fritsch, L., Kohlweiss, M., Kesdogan, D.: Privacy for profitable location based services. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 164–178. Springer, Heidelberg (2005)

[KZSD07]   Kosta, E., Zibuschka, J., Scherner, T., Dumortier, J.: Privacy-enhancing user-friendly identity management for location based services using prime technology — a legal discussion. In: Cyberlaw Security & Private, The Second International Conference on Legal, Security and Privacy Issues in Information Technology (IT), pp. 183–192 (2007)

[KZSD08]   Kosta, E., Zibuschka, J., Scherner, T., Dumortier, J.: Privacy issues in location based services – legal considerations on privacy-enhancing location based services using prime technology. Computer Law & Security Report 24(2), 137–146 (2008)

[LFPR04]   Lindner, T., Fritsch, L., Plank, K., Rannenberg, K.: Exploitation of public and private wifi coverage for new business models. In: Proceedings of the 4th IFIP Conference on E-Commerce, E-Business, and E-Government (I3E), pp. 131–148. Springer, Heidelberg (2004)

[LPC98]   Lazarou, J., Pomeranz, B.H., Corey, P.N.: Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. JAMA (279), 1200–1205 (1998)

[Mic08]   Microsoft. Microsoft healthvault website (2008), http://www.healthvault.com/

[OB05]   Osterberg, L., Blaschke, T.: Adherence to medication. New England journal of medicine (August 2005)

[PH06]   Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management (August 2010), https://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[PIP04]   PIPS Consortium. PIPS - Personalised Information Platforms for Life and Health Services (2004), http://www.pips.eu.org

[Pon04]   Ponemon Institute. The cost of privacy study (2004)

[Pow08]   Powers, J.: Google Health 2018: Worst Case Scenarios (2008), http://www.in3.org/articles/gh2018worst.htm

[PRI04]   Wilikens, M. (ed.) PRIME Work Package 3.1, PRIME – Requirements V1 (2004)

[Rie08]   Ries. EU Green Paper. Technical report, European Parliament (2008)

[Ros99]   Rose, F.: The economics, concept and design of information intermediaries. Physica Verlag, Heidelberg (1999)

[Sab03]   Sabate, E.: Adherence to long-term therapies: evidence for action. Technical report, World Health Organization (2003)

[SL02]   Schackmann, J., Link, H.: Intermediaries for the provision of mass customized digital goods in electronic commerce. Moving into Mass Customization (2002)

[SS07]   Sanna, A., Serafin, R.: User-Controlled Identity Management Systems in Healthcare. World Congress of Innovation and Technology in Healthcare (2007)

[Sta07]   Stange, A.-K.: Datenschutzbewertung von State-of-the-Art Lernmanagementsystemen. Diploma thesis, Technische Universität Dresden (October 2007)

[TK04]   Teltzrow, M., Kobsa, A.: Communication of Privacy and Personalization in E-Business. In: Proceedings of the 1st Workshop "WHOLES: A Multiple View of Individual Privacy in a Networked World", Stockholm, Sweden (2004)

[VR05]   Vakalis, I., Riha, Z.: Airport Security Controls — Application Prototype Specification (2005), http://www.prime-project.eu

[VW04]   Vakalis, I., Willigens, J.-M.: PRIME Application Prototype: Airport controls v.0.23 (October 2004), http://www.prime-project.eu

[WHA98]   Westin, A.F., Harris, L., and Alliances: E-commerce & privacy: What Net Users Want. Technical report, 1998. Conducted for Privacy & American Business and PricewaterhouseCoopers (1998)

[Wor02a]   World Health Organization. The european health report. Technical report, World Health Organization (2002)

[Wor02b]   World Health Organization. The world health report: 2002: Reducing risks, promoting healthy life. Technical report, World Health Organization (2002)

[ZFR⁺07a]   Zibuschka, J., Fritsch, L., Radmacher, M., Scherner, T., Rannenberg, K.: Enabling privacy of real-life lbs: A platform for flexible mobile service provisioning. In: New Approaches for Security, Privacy and Trust in Complex Environments: Proceedings of the 22nd IFIP TC-11 International Information Security Conference, Sandton, South Africa, vol. 232/2007, pp. 325–336 (2007)

[ZFR⁺07b]   Zibuschka, J., Fritsch, L., Radmacher, M., Scherner, T., Rannenberg, K.: Privacy-friendly lbs: A prototype-supported case study. In: Proceedings of the 13th Americas Conference on Information Systems (2007)

[ZSFR06]   Zibuschka, J., Scherner, T., Fritsch, L., Rannenberg, K.: Towards a unified interface for privacy regulation-conformant location-based services. In: W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement (2006)

Part V

Conclusion and Outlook

29

Conclusion and Outlook

Jan Camenisch[1] and Andreas Pfitzmann[2]

[1] IBM Research
[2] TU Dresden

29.1   Conclusion

PRIME has been the first *large scale comprehensive* research project on *user-controlled privacy-enhancing* identity management (IDM).

User-controlled privacy-enhancing means that each single user is put into control w.r.t. his/her PII as much as possible. This is surely better than letting other entities like other users or organizations decide about and being in control of the user's PII. But surely user-controlled privacy only addresses one aspect of privacy, the individual's interest in privacy. Another important aspect of privacy – the social value of privacy, see below – needs further consideration and support.

Comprehensive means (1) bringing diverse research areas (cryptography, system architecture, policies, application design, . . .) and prototypes together, e.g., designing and evaluating early prototypes, learning some lessons how to integrate their achievements, and closing the remaining gaps, (2) striving for minimization of personally identifiable information (PII), e.g., by anonymity of actors and unlinkability of data, and policies (including policy negotiation) ruling how PII may be used, (3) considering policy enforcement as well as policy robustness against change both of the technology base and its security as well as of the evolvement of the regulatory framework at least to some degree. Comprehensive, however, does not mean that all kinds of applications are supported equally well, e.g., community aspects are not really covered within PRIME, at least not at the application level.

Large scale means that system architecture, security and privacy mechanisms, prototypes, terminology, and tutorials are developed, presented to the public and evaluated. It also means having heavy influence on the outside

world, e.g., MS CardSpace, The Liberty Alliance, Higgins, or Sxip's Skipper, just to name a few.

PRIME has successfully shown that the state-of-the-art privacy-enhancing mechanisms can be integrated to form a middleware on top of which applications can be built such that users can assert their rights and take control over their digital private spheres. Our evaluation of the performance of this identity middleware developed shows that its performance even today is quite sufficient for basic applications (e.g., single sign on (SSO), or location-based services (Chapter 25)). But performance issues remain for complex applications, e.g., the collaborative e-learning prototype (CeL, Chapter 24), which have not been taken into account or at least not solved satisfactorily. For such complex applications, either redesign and re-coding of the PRIME-middleware is necessary or one has to wait for more powerful hardware to solve the performance issues.

If we had to do design and build the PRIME middleware again, we would try to earlier detail a complex application, derive its requirements, both w.r.t. functionality and performance, and communicate these detailed requirements to the developers of the IDM middleware. But given the tight limits w.r.t. timing in the execution of such a project, which clearly restricts the lead time of such a detailed requirement analysis, we believe that PRIME has mainly achieved all what was possible within the four years of the project's duration.

Besides building prototypes, PRIME has also put together a social, legal, and economic framework for privacy-enhancing IDM and contributed substantially to the theoretical research in the field. The PRIME Framework (cf. second part of this book) integrates the legal, socio-economic, applications-specific and technical views on privacy-enhancing identity management. It defines the terms and concepts, the problem space, the vision of PRIME, the PRIME solution, application scenarios, and the positioning of PRIME within the landscape of identity management. In terms of research, the PRIME partners have achieved a number of breakthrough results, in many cases making privacy-enhancing suitable for practice. Still, a number of open questions remain which we will discuss in the next section.

29.2   Outlook

29.2.1   Further Research on Identity Management

An impressive body of basic research results related to privacy, trust and identity management were published since David Chaum's seminal publication [Cha81, Cha85] in 1981. In spite of this and the success of large-scale comprehensive research projects such as PRIME many questions are still open or have even not been asked at all. We here point out the major directions, for a

more detailed analysis, we refer to PRIME's annual research reports available from www.prime-project.eu.

In the areas of the individual privacy-enhancing mechanisms ranging from cryptography, policies, to user interfaces a lot of work remains still to be done. That is, PRIME has shown that the basic mechanisms can be readily applied, which however is just a start. Widely employing them still requires substantial theoretical work, in particular, in the areas of user interfaces, policies, and ontologies. In the area of cryptography, more research is needed to make the mechanisms more efficient and practical as well as to invent new mechanisms that enable privacy in new application areas.

In addition, more research on the fundamental limits of IDM is needed: If attribute values of single individuals stand out and are used in several partial identities, these partial identities can, of course, be linked by those attribute values. In this case, IDM cannot really help, but privacy-aware application design is needed to avoid such attributes standing out. So in the mid term, user-controlled privacy-enhancing IDM has to be complemented by *privacy-aware application design.* Corresponding research projects to develop support tools to develop such applications as well as to develop such applications themselves still have to be defined. In addition, engineers as well as policy makers have to be educated about these fundamental limits of privacy-enhancing IDM and how these limits might be overcome.

Finally, Chaum's research was completely focussed on single individuals sharing information with organizations, which mainly was the focus of PRIME as well. Therefore, *multilateral interactions* (Chapter 22), i.e., several individuals interacting with each other as well as with organizations, need further research, both basic and applied. A prominent example of such multilateral interaction scenarios are communities, where appropriate research projects are on their way, see also below.

29.2.2   Making Privacy Real

PRIME has shown that privacy-enhancing identity management is viable and ready to be used in practice. To indeed make PRIME's vision a reality depends on multiple things.

First of all, the technologies as employed by PRIME need to be made available and readily applicable. As these technologies are middleware and infrastructural components, it is particularly hard to deploy them widely. To still achieve this goal, the industry needs to agree on standards that support these technologies and that standardize these technologies. In particular, policies need to be defined so that it can be described and agreed what PII is used for and so that the different components can easily interplay with each other. Besides standards, also open source (reference) implementations need be made available, so that application providers can take advantage of them and build their applications on top. Also, as we are speaking about technologies for security and privacy, having the source code of them public is an essential

part to have users trust these technologies by inspection and verification of the code.

Next, intuitive user interfaces are essential to enable users to easily manage their privacy, identities, and relationships. For all these tasks, these user interfaces have to mediate trust in the technology to the end users. For implementing Privacy and Identity Management tools with configurable, context-dependent and user-controlled attributes, users must have options for configurations and user control, understand them, be aware of when they can be used and be empowered (i.e., be able to easily understand how) to use them to exercise control. Intuitive user interfaces shall promote legal privacy principles, i.e., user interfaces that enforce these HCI requirements of user comprehension, consciousness and control.

Finally, educational material needs to be provided:

- for end-users to understand the problems and to learn how they can address them;
- for application designers, engineers, and suppliers so that they know how to design and build privacy-aware solutions; and
- for policy and decision makers so that they become aware of the possibilities of these privacy-enhancing mechanisms, can guide the public discussions, and provide the necessary framework for privacy, trust, and identity management.

29.2.3   Including the Social Value of Privacy

Privacy both has an individual value and a social value: The *individual value of privacy* is that all choice and decision is given to the individual, who decides about how to live. The *social value of privacy* stems from the needs of democratic societies which only can function if citizens speak out what the really want. But if citizens feel that there might be surveillance, i.e., citizens are not assured of sufficiently strong privacy, then citizens do not say what they want, but what they assume the majority wants, making society less innovative, less stable and much more prone to manipulation of various kinds.

PRIME – as each research project on user-controlled privacy – mainly addresses the individual value of privacy and nearly not the social value, i.e., within the limits of law, all choice is given to the individuals. However, society might require that users keep privacy for reasons of democracy.

Within PRIME, discussions of business models similarly are about whether individuals or companies would pay for privacy of individuals. But even if neither would be willing to pay, society would want citizens to have privacy.

Therefore, even if it is unclear today who is going to pay, this means a positive message related to the economy of privacy-enhancing technologies (PETs) within democratic societies: Someone is going to pay for PETs.

In the past and at present, data protection regulations only cover organizations, since in the past it has been only organizations who had very powerful

means of data processing. At present and for the future, this turns out to have changed: Even individuals of moderate income have data processing capabilities available very large organizations dreamed of twenty years ago. Therefore, the question arises: Should the legal framework include private data processing and what would be the consequences? We actually think that data processing should all be treated the same as organizations and individuals all have the same kind of technology available. Of course we see that there might be different limits of data protection regulations addressing organizations as data processors than addressing individuals as data processors. But ignoring individuals as data processors w.r.t. regulation quite probably is not a viable approach for the future.

Another regulatory issue is that there are lots of issues that technologies as developed by PRIME cannot address. Pseudonomous profiling, e.g., as done by Google and Doubleclick falls through legislation and cannot be prevented (by technology and local regulations, nor other pets). The global regulatory framework has to deal with such processing of data which can become personal data very easily by linking them to other data, e.g., after a merger of companies or after a security breach in another company.

For the future, the main challenge is bringing user-controlled privacy and organization-controlled privacy (which both are driven by aims of individual actors, be it individuals, be it organizations), which are mainly implemented by action, ICT, and money, together with society-controlled privacy which is driven by regulation. How to make action, ICT, money *and* regulation play together to suit the needs of individuals, organizations and democratic society at large?

29.2.4 Succeeding PRIME

Regarding privacy, trust, and identity management, some projects have been defined that pick up the work where PRIME has stopped. We mentioned the two that involve PRIME partners and hence can be regarded as successors of PRIME.

PRIME has shown that privacy technologies can enable citizens to execute their legal rights to control personal information in on-line transactions. Now, the increasingly collaborative character of the Internet enables anyone to compose services and contribute and distribute information. Individuals will contribute throughout their lives leaving life-long trails of personal data. This raises substantial new privacy challenges: A first technical challenge is how to protect privacy in emerging Internet applications such as collaborative scenarios and virtual communities. A second challenge is how to maintain life-long privacy.

PrimeLife will resolve the core privacy and trust issues pertaining to these challenges. Its long-term vision is to counter the trend to life-long personal data trails without compromising on functionality. PrimeLife will build upon

and expand the sound foundation laid by PRIME. See `www.primelife.eu` for more information.

Online communities connect millions of people around the world, to communicate, interact and share interests. As these communities are getting more mobile, becoming a ubiquitous part of our lives, new opportunities and challenges emerge.

PICOS has the mission to investigate mobile communities and their services. Especially regarding aspects like privacy and identity management as well as technical and economic aspects. See `www.picos-project.eu`.

References

[Cha81] Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24(2), 84–88 (1981)

[Cha85] Chaum, D.: Security without identification: Card computers to make big brother obsolete. Communications of the ACM 28(10), 1030–1044 (1985)

Part VI

Appendix

XML Schemata

30.1 Access Control and Release Language: XML Schema

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:Q1="http://www.prime-project.eu/policies/"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
 elementFormDefault="qualified"
 targetNamespace="http://www.prime-project.eu/policies/"
 xmlns:Q2="https://www.prime-project.eu/ont/XSD-Claim"
 xmlns:request="https://www.prime-project.eu/ont/XSD-ClaimRequest">
 <xs:import namespace="https://www.prime-project.eu/ont/XSD-ClaimRequest"
   schemaLocation="ClaimRequest.xsd" />
   <xs:element name="policy" type="Q1:policyType">
     <xs:annotation>
       <xs:documentation>One policy.</xs:documentation>
     </xs:annotation>
   </xs:element>
 <xs:complexType name="policyType">
   <xs:sequence>
     <xs:element maxOccurs="1" minOccurs="1" name="description"
      type="Q1:descriptionType" />
      <xs:element name="object" type="xs:anyURI">
        <xs:annotation>
          <xs:documentation>What to access.</xs:documentation>
        </xs:annotation>
      </xs:element>
     <xs:element name="objectExprs" type="Q1:conditionListType" />
     <xs:element name="actions" type="Q1:actionListType">
       <xs:annotation>
         <xs:documentation>Allowed actions.</xs:documentation>
       </xs:annotation>
     </xs:element>
     <xs:element name="rules" type="Q1:ruleListType" />
   </xs:sequence>
   <xs:attribute name="id" type="xs:ID">
     <xs:annotation>
       <xs:documentation>Id of the policy.</xs:documentation>
     </xs:annotation>
   </xs:attribute>
   <xs:attribute name="type">
     <xs:annotation>
       <xs:documentation>
         A policy either guarding the disclose of or access to data.
       </xs:documentation>
     </xs:annotation>
     <xs:simpleType>
```

```xml
      <xs:restriction base="xs:anyURI">
        <xs:enumeration value="http://www.prime-project.eu/policies/access" />
        <xs:enumeration value="http://www.prime-project.eu/policies/release" />
        <xs:enumeration value="http://www.prime-project.eu/policies/datahandling" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="authors" type="xs:string" />
</xs:complexType>
<xs:complexType name="ruleListType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="rule" type="Q1:ruleType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ruleType">
  <xs:sequence>
    <xs:element name="subject" type="xs:anyURI" />
    <xs:element name="actions" type="Q1:actionListType" />
    <xs:element name="purposes" type="Q1:purposesListType" />
    <xs:element name="conditions" type="Q1:expressionsType" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required" />
</xs:complexType>
<xs:complexType name="groupListType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="group" type="Q1:group" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="expressionsType">
  <xs:sequence>
    <xs:element name="subjectExprs" type="Q1:groupListType" />
    <xs:element name="genericExprs" type="Q1:conditionListType" />
    <xs:element name="trustExprs" type="Q1:conditionListType" />
    <xs:element name="lbsExprs" type="Q1:conditionListType" />
    <xs:element name="stateExprs" type="Q1:groupListType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="conditionType">
  <xs:complexContent>
    <xs:extension base="Q1:predicateType">
      <xs:attribute default="true" name="sanitization" type="xs:boolean" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="group">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" name="condition"
     type="Q1:conditionType" />
    <xs:element maxOccurs="1" minOccurs="1" name="evidence"
     type="request:claimrequest.option.group.evidenceListType" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:anyURI" use="required" />
</xs:complexType>
<xs:complexType name="annotationType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attribute name="name" type="xs:anyURI" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="descriptionType">
  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="long" type="xs:string" />
    <xs:element name="annotation" type="Q1:annotationType" />
  </xs:sequence>
  <xs:attribute name="short" type="xs:string" />
</xs:complexType>
```

```
<xs:complexType name="conditionListType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="condition"
     type="Q1:conditionType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="actionListType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" name="action"
     type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="predicateType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="argument"
     type="Q1:argumentType" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:anyURI" />
</xs:complexType>
<xs:complexType name="certificationType">
  <xs:sequence>
    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##any"
     processContents="lax" />
  </xs:sequence>
  <xs:attribute name="type" type="xs:anyURI" />
</xs:complexType>
<xs:complexType name="argumentType">
  <xs:complexContent>
    <xs:extension base="xs:anyType">
      <xs:attribute name="isLiteral" type="xs:boolean" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="purposesListType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" name="purpose"
     type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="recipientsListType">
  <xs:sequence>
    <xs:element name="recipient" type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

30.2   Data Handling Language: XML Schema

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
 attributeFormDefault="unqualified">
  <xs:complexType name="condition">
    <xs:sequence>
      <xs:element name="argument" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="xs:anyType">
              <xs:attribute name="isLiteral" type="xs:boolean" />
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="xs:anyURI" use="required" />
```

```
            </xs:complexType>
            <xs:element name="DHP">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="annotation">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="description" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="Description">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Long" type="xs:string" />
                        <xs:element name="Short" type="xs:string" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="PII" type="xs:anyURI" />
                  <xs:element name="actions">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="action" type="xs:anyURI" minOccurs="0"
                         maxOccurs="unbounded" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="purposes">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="purpose" minOccurs="0" maxOccurs="unbounded">
                          <xs:complexType>
                            <xs:simpleContent>
                              <xs:extension base="xs:string">
                                <xs:attribute name="name" type="xs:anyURI" use="required" />
                                <xs:attribute name="type" use="optional" default="optional">
                                  <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                      <xs:enumeration value="required" />
                                      <xs:enumeration value="optional" />
                                    </xs:restriction>
                                  </xs:simpleType>
                                </xs:attribute>
                              </xs:extension>
                            </xs:simpleContent>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="recipients">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="recipient" maxOccurs="unbounded">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element name="condition" type="condition" minOccurs="0"
                               maxOccurs="unbounded" />
                            </xs:sequence>
                            <xs:attribute name="subject" type="xs:anyURI" />
                            <xs:attribute name="type" use="optional" default="optional">
                              <xs:simpleType>
                                <xs:restriction base="xs:string">
                                  <xs:enumeration value="required" />
                                  <xs:enumeration value="optional" />
                                </xs:restriction>
```

```
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="obligations">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="condition" type="condition" minOccurs="0"
        maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="provisions">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="condition" type="condition" minOccurs="0"
        maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="gen_conditions">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="condition" type="condition" minOccurs="0"
        maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="disputes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="dispute" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="remedies" type="xs:string" />
              <xs:element name="description" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="thirdParty" type="xs:anyURI" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="id" type="xs:string" />
</xs:complexType>
</xs:element>
</xs:schema>
```

Author Index