Steven Furnell
Sokratis K. Katsikas
Antonio Lioy (Eds.)

LNCS 5185

Trust, Privacy and Security in Digital Business

5th International Conference, TrustBus 2008
Turin, Italy, September 2008
Proceedings

Springer

Lecture Notes in Computer Science 5185

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Steven Furnell   Sokratis K. Katsikas
Antonio Lioy (Eds.)

Trust, Privacy and Security in Digital Business

5th International Conference, TrustBus 2008
Turin, Italy, September 4-5, 2008
Proceedings

Springer

Volume Editors

Steven Furnell
University of Plymouth
School of Computing, Communications and Electronics
A310, Portland Square, Drake Circus, Plymouth, Devon PL4 8AA, UK
E-mail: sfurnell@jack.see.plymouth.ac.uk

Sokratis K. Katsikas
University of Piraeus
Department of Technology Education and Digital Systems
150 Androutsou St., 18534 Piraeus, Greece
E-mail: ska@unipi.gr

Antonio Lioy
Politecnico di Torino
Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
E-mail: lioy@polito.it

Library of Congress Control Number: 2008933371

CR Subject Classification (1998): K.4.4, K.4, K.6, E.3, C.2, D.4.6, J.1

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-85734-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-85734-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12511266 06/3180 5 4 3 2 1 0

Preface

This book contains the proceedings of the 5th International Conference on Trust, Privacy and Security in Digital Business (TrustBus 2008), held in Turin, Italy on 4–5 September 2008. Previous events in the TrustBus series were held in Zaragoza, Spain (2004), Copenhagen, Denmark (2005), Krakow, Poland (2006), and Regensburg, Germany (2007). TrustBus 2008 brought together academic researchers and industrial developers to discuss the state of the art in technology for establishing trust, privacy and security in digital business. We thank the attendees for coming to Turin to participate and debate upon the latest advances in this area.

The conference program included one keynote presentation and six technical paper sessions. The keynote speech was delivered by Andreas Pfitzmann from the Technical University of Dresden, Germany, on the topic of "Biometrics – How to Put to Use and How Not at All". The reviewed paper sessions covered a broad range of topics, including trust and reputation systems, security policies and identity management, privacy, intrusion detection and authentication, authorization and access control. Each of the submitted papers was assigned to five referees for review. The program committee ultimately accepted 18 papers for inclusion in the proceedings.

We would like to express our thanks to the various people who assisted us in organizing the event and formulating the program. We are very grateful to the program committee members and the external reviewers for their timely and thorough reviews of the papers. Thanks are also due to the DEXA organizing committee for supporting our event, and in particular to Gabriela Wagner for her assistance and support with the administrative aspects.

Finally we would like to thank all the authors that submitted papers for the event, and contributed to an interesting set of conference proceedings.

September 2008

Steven Furnell
Sokratis Katsikas
Antonio Lioy

Organization

Program Committee

General Chairperson

Antonio Lioy	Politecnico di Torino, Italy

Conference Program Chairpersons

Steven Furnell,	University of Plymouth, UK
Sokratis Katsikas	University of Piraeus, Greece

Program Committee Members

Vijay Atluri	Rutgers University, USA
Marco Casassa Mont	HP Labs Bristol, UK
David Chadwick	University of Kent, UK
Nathan Clarke	University of Plymouth, UK
Richard Clayton	University of Cambridge, UK
Frederic Cuppens	ENST Bretagne, France
Ernesto Damiani	Università degli Studi di Milano, Italy
Ed Dawson	Queensland University of Technology, Australia
Sabrina De Capitani di Vimercati	University of Milan, Italy
Hermann De Meer	University of Passau, Germany
Jan Eloff	University of Pretoria, South Africa
Eduardo B. Fernandez	Florida Atlantic University, USA
Carmen Fernandez-Gago	University of Malaga, Spain
Elena Ferrari	University of Insubria, Italy
Simone Fischer-Huebner	University of Karlstad, Sweden
Carlos Flavian	University of Zaragoza, Spain
Juan M. Gonzalez-Nieto	Queensland University of Technology, Australia
Rüdiger Grimm	University of Koblenz, Germany
Dimitris Gritzalis	Athens University of Economics and Business, Greece
Stefanos Gritzalis	University of the Aegean, Greece
Ehud Gudes	Ben-Gurion University, Israel
Sigrid Gürgens	Fraunhofer Institute for Secure Information Technology, Germany
Carlos Gutierrez	University of Castilla-La Mancha, Spain

Marit Hansen	Independent Center for Privacy Protection, Germany
Audun Jøsang	Queensland University of Technology, Australia
Tom Karygiannis	NIST, USA
Dogan Kesdogan	NTNU Trondheim, Norway
Hiroaki Kikuchi	Tokai University, Japan
Spyros Kokolakis	University of the Aegean, Greece
Costas Lambrinoudakis	University of the Aegean, Greece
Leszek Lilien	Western Michigan University, USA
Javier Lopez	University of Malaga, Spain
Antonio Mana Gomez	University of Malaga, Spain
Olivier Markowitch	Université Libre de Bruxelles, Belgium
Fabio Martinelli	CNR, Italy
Chris Mitchell	Royal Holloway College, University of London, UK
Guenter Mueller	University of Freiburg, Germany
Eiji Okamoto	University of Tsukuba, Japan
Martin S. Olivier	University of Pretoria, South Africa
Rolf Oppliger	eSecurity Technologies, Switzerland
Maria Papadaki	University of Plymouth, UK
Ahmed Patel	Kingston University, UK
Guenther Pernul	University of Regensburg, Germany
Andreas Pfitzmann	Dresden University of Technology, Germany
Hartmut Pohl	FH Bonn-Rhein-Sieg, Germany
Karl Posch	University of Technology Graz, Austria
Torsten Priebe	Capgemini, Austria
Gerald Quirchmayr	University of Vienna, Austria
Christoph Ruland	University of Siegen, Germany
Pierangela Samarati	University of Milan, Italy
Matthias Schunter	IBM Zurich Research Lab., Switzerland
Mikko T. Siponen	University of Oulu, Finland
Adrian Spalka	CompuGROUP Holding AG, Germany
A Min Tjoa	Technical University of Vienna, Austria
Allan Tomlinson	Royal Holloway College, University of London, UK
Christos Xenakis	University of Piraeus, Greece
Jianying Zhou	I2R, Singapore

External Reviewers

Carlos A. Gutierrez Garcia	University of Castilla-La Mancha, Spain
Andrea Perego	University of Insubria, Italy

Table of Contents

Invited Lecture

Trust

Authentication, Authorization and Access Control

Reputation Systems

Security Policies and Identity Management

Intrusion Detection and Applications of Game Theory to IT Security Problems

Privacy

Biometrics – How to Put to Use and How Not at All?

Andreas Pfitzmann

TU Dresden, Faculty of Computer Science, 01062 Dresden, Germany
Andreas.Pfitzmann@tu-dresden.de

Abstract. After a short introduction to biometrics w.r.t. IT security, we derive conclusions on how biometrics should be put to use and how not at all. In particular, we show how to handle security problems of biometrics and how to handle security and privacy problems caused by biometrics in an appropriate way. The main conclusion is that biometrics should be used between human being and his/her personal devices only.

1 Introduction

Biometrics is advocated as *the* solution to admission control nowadays. But what can biometrics achieve, what not, which side effects do biometrics cause and which challenges in system design do emerge?

1.1 What Is Biometrics?

Measuring physiological or behavioral characteristics of persons is called biometrics. Measures include the *physiological characteristics*

- (shape of) face,
- facial thermograms,
- fingerprint,
- hand geometry,
- vein patterns of the retina,
- patterns of the iris, and
- DNA

and the *behavioral characteristics*

- dynamics of handwriting (e.g., handwritten signatures),
- voice print, and
- gait.

One might make a distinction whether the person whose physiological or behavioral characteristics are measured has to participate explicitly (*active* biometrics), so (s)he gets to know that a measurement takes place, or whether his/her explicit participation is not necessary (*passive* biometrics), so (s)he might not notice that a measurement takes place.

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 1–7, 2008.

1.2 Biometrics for What Purpose?

Physiological or behavioral characteristics are measured and compared with reference values to

Authenticate (Is this the person (s)he claims to be?), or even to
Identify (Who is this person?).

Both decision problems are the more difficult the larger the set of persons of which individual persons have to be authenticated or even identified. Particularly in the case of identification, the precision of the decision degrades with the number of possible persons drastically.

2 Security Problems of Biometrics

As with all decision problems, biometric authentication/identification may produce two kinds of errors [1]:

False nonmatch rate: Persons are wrongly not authenticated or wrongly not identified.
False match rate: Persons are wrongly authenticated or wrongly identified.

False nonmatch rate and false match rate can be traded off by adjusting the decision threshold. Practical experience has shown that only one error rate can be kept reasonably small – at the price of a unreasonably high error rate for the other type.

A biometric technique is more secure for a certain application area than another biometric technique if both error types occur more rarely. It is possible to adapt the threshold of similarity tests used in biometrics to various application areas. But if only one of the two error rates should be minimized to a level that can be provided by well managed authentication and identification systems that are based on people's knowledge (e.g., passphrase) or possession (e.g., chip card), today's biometric techniques can only provide an unacceptably high error rate for the other error rate.

Since more than two decades we hear announcements that biometric research will change this within two years or within four years at the latest. In the meantime, I doubt whether such a biometric technique exists, if the additional features promised by advocates of biometrics shall be provided as well:

- user-friendliness, which limits the quality of data available to pattern recognition, and
- acceptable cost despite possible attackers who profit from technical progress as well (see below).

In addition to this decision problem being an inherent security problem of biometrics, the implementation of biometric authentication/identification has to ensure that the biometric data come from the person at the time of verification and are neither replayed in time nor relayed in space [2]. This may be more difficult than it sounds, but it is a common problem of all authentication/identification mechanisms.

3 Security Problems Caused by Biometrics

Biometrics does not only have the security problems sketched above, but the use of biometrics also creates new security problems. Examples are given in the following.

3.1 Devaluation of Classic Forensic Techniques Compromises Overall Security

Widespread use of biometrics can devaluate classic forensic techniques – as sketched for the example of fingerprints – as a means to trace people and provide evidence:

Databases of fingerprints or common issuing of one's fingerprint essentially ease the fabrication of finger replicas [3] and thus leaving someone else's fingerprints at the site of crime. And the more fingerprints a forger has at his discretion and the more he knows about the holder of the fingerprints, the higher the plausibility of somebody else's fingerprints he will leave. Plausible fingerprints at the site of crime will cause police or secret service at least to waste time and money in their investigations – if not to accuse the wrong suspects in the end.

If biometrics based on fingerprints is used to secure huge values, quite probably, an "industry" fabricating replicas of fingers will arise. And if fingerprint biometrics is rolled out to the mass market, huge values to be secured arise by accumulation automatically. It is unclear whether society would be well advised to try to ban that new "industry" completely, because police and secret services will need its services to gain access to, e.g., laptops secured by fingerprint readers (assuming both the biometrics within the laptops and the overall security of the laptops get essentially better than today). Accused people may not be forced to co-operate to overcome the barrier of biometrics at their devices at least under some jurisdictions. E.g., according to the German constitution, nobody can be forced to co-operate in producing evidence against himself or against close relatives.

As infrastructures, e.g., for border control, cannot be upgraded as fast as single machines (in the hands of the attackers) to fabricate replicas of fingers, a loss of security is to be expected overall.

3.2 Stealing Body Parts (Safety Problem of Biometrics)

In the press you could read that one finger of the driver of a Mercedes S-class has been cut off to steal his car [4]. Whether this story is true or not, it does exemplify a problem I call the safety problem of biometrics:

- Even a temporary (or only assumed) improvement of "security" by biometrics is not necessarily an advance, but endangers physical integrity of persons.
- If checking that the body part measured biometrically is still alive really works, kidnapping and blackmailing will replace the stealing of body parts.

If we assume that as a modification of the press story, the thieves of the car know they need the finger as part of a functioning body, they will kidnap the owner of the car and take him and the car with them to a place where they will remove the biometric security from the car. Since such a place usually is closely connected to the thieves and probably gets to be known by the owner of the car, they will probably kill the owner after arriving at that place to protect their identities. So biometrics checking that the measured body part of a person is still alive may not solve the safety problem, but exacerbate it.

3.3 Favored Multiple Identities Could Be Uncovered as Well

The naive dream of politicians dealing with public safety to recognize or even identify people by biometrics unambiguously will become a nightmare if we do not completely ignore that our societies need multiple identities. They are accepted and often useful for agents of secret services, undercover agents, and persons in witness-protection programs.

The effects of a widespread use of biometrics would be:

- To help uncover agents of secret services, each country will set up person-related biometric databases at least for all foreign citizens.
- To help uncover undercover agents and persons in witness-protection programs, in particular organized crime will set up person-related biometric databases.

Whoever believes in the success of biometric authentication and identification, should *not* employ it on a large scale, e.g., in passports.

4 Privacy Problems Caused by Biometrics

Biometrics is not only causing security problems, but privacy problems as well:

1. Each biometric measurement contains potentially sensitive personal data, e.g., a retina scan reveals information on consumption of alcohol during the last two days, and it is under discussion, whether fingerprints reveal data on homosexuality [5,6].
2. Some biometric measurements might take place (passive biometrics) without knowledge of the data subject, e.g., (shape of) face recognition.

In practice, the security problems of biometrics will exacerbate their privacy problems:

3. Employing several kinds of biometrics in parallel, to cope with the insecurity of each single kind [7], multiplies the privacy problems (cf. mosaic theory of data protection).

Please take note of the principle that data protection by erasing personal data does not work, e.g., on the Internet, since it is necessary to erase *all* copies. Therefore even the possibility to gather personal data has to be avoided. This means: no biometric measurement.

5 How to Put to Use and How Not at All?

Especially because biometrics has security problems itself and additionally can cause security and privacy problems, one has to ask the question how biometrics should be used and how it should not be used at all.

5.1 Between Data Subject and His/Her Devices

Despite the shortcomings of current biometric techniques, if adjusted to low false nonmatch rates, they can be used between a human being and his/her personal devices. This is even true if biometric techniques are too insecure to be used in other applications or cause severe privacy or security problems there:

- Authentication by possession and/or knowledge *and* biometrics improves security of authentication.
- No devaluation of classic forensic techniques, since the biometric measurements by no means leave the device of the person and persons are not conditioned to divulge biometric features to third-party devices.
- No privacy problems caused by biometrics, since each person (hopefully) is and stays in control of his/her devices.
- The safety problem of biometrics remains unchanged. But if a possibility to switch off biometrics completely and forever after successful biometric authentication is provided and this is well known to everybody, then biometrics does not endanger physical integrity of persons, if users are willing to cooperate with determined attackers. Depending on the application context of biometrics, compromises between no possibility at all to disable biometrics and the possibility to completely and permanently disable biometrics might be appropriate.

5.2 Not at All between Data Subject and Third-Party Devices

Regrettably, it is to be expected that it will be tried to employ biometrics in other ways, i.e. between human being and third-party devices. This can be done using active or passive biometrics:

- Active biometrics in passports and/or towards third-party devices is noted by the person. This helps him/her to avoid active biometrics.
- Passive biometrics by third-party devices cannot be prevented by the data subjects themselves – regrettably. Therefore, at least *covertly employed passive biometrics should be forbidden by law.*

What does this mean in a world where several countries with different legal systems and security interests (and usually with no regard of foreigners' privacy) accept entry of foreigners into their country only if the foreigner's country issued a passport with machine readable and testable digital biometric data or the foreigner holds a stand-alone visa document containing such data?

5.3 Stand-Alone Visas Including Biometrics or Passports Including Biometrics?

Stand-alone visas including biometrics do much less endanger privacy than passports including biometrics. This is true both w.r.t. foreign countries as well as w.r.t. organized crime:

- Foreign countries will try to build up person-related biometric databases of visitors – we should not ease it for them by conditioning our citizens to accept biometrics nor should we make it cheaper for them by including machine-readable biometrics in our passports.
- Organized crime will try to build up person-related biometric databases – we should not ease it for them by establishing it as common practice to deliver biometric data to third-party devices, nor should we help them by making our passports machine readable without keeping the passport holder in control[1]

Since biometric identification is all but perfect, different measurements and thereby different values of biometric characteristics are less suited to become a universal personal identifier than a digital reference value constant for 10 years in your passport. Of course this only holds if these different values of biometric characteristics are not always "accompanied" by a constant universal personal identifier, e.g., the passport number.

Therefore, countries taking privacy of their citizens seriously should

- not include biometric characteristics in their passports or at least minimize biometrics there, and
- mutually agree to issue – if heavy use of biometrics, e.g., for border control, is deemed necessary – stand-alone visas including biometric characteristics, but not to include any data usable as a universal personal identifier in these visas, nor to gather such data in the process of issuing the visas.

6 Conclusions

Like the use of every security mechanism, the use of biometrics needs circumspection and possibly utmost caution. In any case, in democratic countries the widespread use of biometrics in passports needs a qualified and manifold debate. This debate took place at most partially and unfortunately it is not encouraged by politicians dealing with domestic security in the western countries. Some politicians even refused it or – if this has not been possible – manipulated the debate by making indefensible promises or giving biased information.

This text shows embezzled or unknown arguments regarding biometrics und tries to contribute to a qualified and manifold debate on the use of biometrics.

[1] cf. insecurity of RFID-chips against unauthorized reading, http://dud.inf.tu-dresden.de/literatur/Duesseldorf2005.10.27Biometrics.pdf

7 Outlook

After a discussion on how to balance domestic security and privacy, an investigation of authentication and identification infrastructures [8] that are able to implement this balance should start:

- Balancing surveillance and privacy should not only happen concerning single applications (e.g. telephony, e-mail, payment systems, remote video monitoring), but across applications.
- Genome databases, which will be built up to improve medical treatment in a few decades, will possibly undermine the security of biometrics which are predictable from these data.
- Genome databases and ubiquitous computing (= pervasive computing = networked computers in all physical things) will undermine privacy primarily in the physical world – we will leave biological or digital traces wherever we are.
- Privacy spaces in the digital world are possible (and needed) and should be established – instead of trying to gather and store traffic data for a longer period of time at high costs and for (very) limited use (in the sense of balancing across applications).

Acknowledgements

Many thanks to my colleagues in general and Rainer Böhme, Katrin Borcea-Pfitzmann, Dr.-Ing. Sebastian Clauß, Marit Hansen, Matthias Kirchner, and Sandra Steinbrecher in particular for suggestions to improve this paper and some technical support.

References

1. Jain, A., Hong, L., Pankanti, S.: Biometric Identification. Communications of the ACM 43/2, 91–98 (2000)
2. Schneier, B.: The Uses and Abuses of Biometrics. Communications of the ACM 42/8, 136 (1999)
3. Chaos Computer Club e.V.: How to fake fingerprints? (June 12, 2008), http://www.ccc.de/biometrie/fingerabdruck_kopieren.xml?language=en
4. Kent, J.: Malaysia car thieves steal finger (June 16, 2008), news.bbc.co.uk/2/hi/asia-pacific/4396831.stm
5. Hall, J.A.Y., Kimura, D.: Dermatoglyphic Asymmetry and Sexual Orientation in Men. Behavioral Neuroscience 108, 1203–1206 (1994) (June 12, 2008), www.sfu.ca/~dkimura/articles/derm.htm
6. Forastieri, V.: Evidence against a Relationship between Dermatoglyphic Asymmetry and Male Sexual Orientation. Human Biology 74/6, 861–870 (2002)
7. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer, New York (2006)
8. Pfitzmann, A.: Wird Biometrie die IT-Sicherheitsdebatte vor neue Herausforderungen stellen? DuD, Datenschutz und Datensicherheit, Vieweg-Verlag 29/5, 286–289 (2005)

A Map of Trust between Trading Partners

John Debenham[1] and Carles Sierra[2]

[1] University of Technology, Sydney, Australia
debenham@it.uts.edu.au
[2] Institut d'Investigacio en Intel.ligencia Artificial, Spanish Scientific Research Council, UAB
08193 Bellaterra, Catalonia, Spain
sierra@iiia.csic.es

Abstract. A pair of 'trust maps' give a fine-grained view of an agent's accumulated, time-discounted belief that the enactment of commitments by another agent will be in-line with what was promised, and that the observed agent will act in a way that respects the confidentiality of previously passed information. The structure of these maps is defined in terms of a categorisation of utterances and the ontology. Various summary measures are then applied to these maps to give a succinct view of trust.

1 Introduction

The intuition here is that trust between two trading partners is derived by observing two types of behaviour. First, an agent exhibits trustworthy behaviour through the enactment of his commitments being in-line with what was promised, and second, it exhibits trustworthy behaviour by respecting the confidentiality of information passed 'in confidence'. Our agent observes both of these types of behaviour in another agent and represents each of them on a map. The structure of these two maps is defined in terms of both the type of behaviour observed and the ontology. The first 'map' of trust represents our agent's accumulated, time-discounted belief that the enactment of commitments will be in-line with what was promised. The second map represents our agent's accumulated, time-discounted belief that the observed agent will act in a way that fails to respect the confidentiality of previously passed information.

The only action that a software agent can perform is to send an utterance to another agent. So trust, and any other high-level description of behaviour, must be derived by observing this act of message passing. We use the term *private information* to refer to anything that one agent knows that is not known to the other. The intention of transmitting any utterance should be to convey some private information to the receiver — otherwise the communication is worthless. In this sense, trust is built through exchanging, and subsequently validating, private information [1]. Trust is seen in a broad sense as a measure of the strength of the relationship between two agents, where the *relationship* is the history of the utterances exchanged. To achieve this we categorise utterances as having a particular type and by reference to the ontology — this provides the structure for our map.

The literature on trust is enormous. The seminal paper [2] describe two approaches to trust: first, as a belief that another agent will do what it says it will, or will reciprocate

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 8–17, 2008.

for common good, and second, as constraints on the behaviour of agents to conform to trustworthy behaviour. The map described here is concerned with the first approach where trust is something that is learned and evolves, although this does not mean that we view the second as less important [3]. The map also includes reputation [4] that feeds into trust. [5] presents a comprehensive categorisation of trust research: policy-based, reputation-based, general *and* trust in information resources — for our trust maps, the estimating the integrity of information sources is fundamental. [6] presents an interesting taxonomy of trust models in terms of nine types of trust model. The scope described there fits well within the map described here with the possible exception of identity trust and security trust. [7] describes a powerful model that integrates interaction an role-based trust with witness and certified reputation that also relate closely to our model.

A key aspect of the behaviour of trading partners is the way in which they enact their commitments. The enactment of a contract is uncertain to some extent, and trust, precisely, is a measure of how uncertain the enactment of a contract is. Trust is therefore a *measure of expected deviations of behaviour* along a dimension determined by the type of the contract. A unified model of trust, reliability and reputation is described for a breed of agents that are grounded on information-based concepts [8]. This is in contrast with previous work that has focused on the similarity of offers [9,10], game theory [11], or first-order logic [12].

We assume that a multiagent system $\{\alpha, \beta_1, \ldots, \beta_o, \xi, \theta_1, \ldots, \theta_t\}$, contains an agent α that interacts with negotiating agents, β_i, information providing agents, θ_j, and an *institutional agent*, ξ, that represents the institution where we assume the interactions happen [3]. Institutions provide a normative context that simplifies interaction. We understand agents as being built on top of two basic functionalities. First, a *proactive machinery*, that transforms *needs* into *goals* and these into *plans* composed of *actions*. Second, a reactive machinery, that uses the received messages to obtain a new world model by updating the probability distributions in it.

2 Ontology

In order to define a language to structure agent dialogues we need an ontology that includes a (minimum) repertoire of elements: a set of *concepts* (e.g. quantity, quality, material) organised in a is-a hierarchy (e.g. platypus is a mammal, Australian-dollar is a currency), and a set of relations over these concepts (e.g. price(beer,AUD)).[1] We model ontologies following an algebraic approach as:

An ontology is a tuple $O = (C, R, \leq, \sigma)$ where:

1. C is a finite set of concept symbols (including basic data types);
2. R is a finite set of relation symbols;
3. $\leq$ is a reflexive, transitive and anti-symmetric relation on C (a partial order)
4. $\sigma : R \to C^+$ is the function assigning to each relation symbol its arity

where $\leq$ is the traditional *is-a* hierarchy. To simplify computations in the computing of probability distributions we assume that there is a number of disjoint *is-a* trees covering

[1] Usually, a set of axioms defined over the concepts and relations is also required. We will omit this here.

different ontological spaces (e.g. a tree for types of fabric, a tree for shapes of clothing, and so on). R contains relations between the concepts in the hierarchy, this is needed to define 'objects' (e.g. deals) that are defined as a tuple of issues.

The semantic distance between concepts within an ontology depends on how far away they are in the structure defined by the $\leq$ relation. Semantic distance plays a fundamental role in strategies for information-based agency. How signed contracts, $\mathit{Commit}(\cdot)$, about objects in a particular semantic region, and their execution, $\mathit{Done}(\cdot)$, *affect* our decision making process about signing future contracts in nearby semantic regions is crucial to modelling the common sense that human beings apply in managing trading relationships. A measure [13] bases the *semantic similarity* between two concepts on the *path length* induced by $\leq$ (more distance in the $\leq$ graph means less semantic similarity), and the *depth* of the subsumer concept (common ancestor) in the shortest path between the two concepts (the deeper in the hierarchy, the closer the meaning of the concepts). Semantic similarity is then defined as:

$$\delta(c,c') = e^{-\kappa_1 l} \cdot \frac{e^{\kappa_2 h} - e^{-\kappa_2 h}}{e^{\kappa_2 h} + e^{-\kappa_2 h}}$$

where l is the length (i.e. number of hops) of the shortest path between the concepts c and c', h is the depth of the deepest concept subsuming both concepts, and κ_1 and κ_2 are parameters scaling the contributions of the shortest path length and the depth respectively.

3 Doing the 'Right Thing'

We now describe our first 'map' of the trust that represents our agent's accumulated, time-discounted belief that the enactment of commitments by another agent will be in-line with what was promised. This description is fairly convoluted. This sense of trust is built by continually observing the discrepancies, if any, between promise and enactment. So we describe:

1. How an utterance is represented in, and so changes, the world model.
2. How to estimate the 'reliability' of an utterance — this is required for the previous step.
3. How to measure the agent's accumulated evidence.
4. How to represent the measures of evidence on the map.

3.1 Updating the World Model

α's world model consists of probability distributions that represent its uncertainty in the world's state. α is interested in the degree to which an utterance accurately describes what will subsequently be observed. All observations about the world are received as utterances from an all-truthful institution agent ξ. For example, if β communicates the goal "I am hungry" and the subsequent negotiation terminates with β purchasing a book from α (by ξ advising α that a certain amount of money has been credited to α's account) then α may conclude that the goal that β chose to satisfy was something other than hunger. So, α's world model contains probability distributions that represent its uncertain expectations of what will be observed on the basis of utterances received.

We represent the relationship between *utterance*, φ, and subsequent *observation*, φ', in the world model $\mathcal{M}^t$ by $\mathbb{P}^t(\varphi'|\varphi) \in \mathcal{M}^t$, where φ' and φ may be expressed in terms of ontological categories in the interest of computational feasibility. For example, if φ is "I will deliver a bucket of fish to you tomorrow" then the distribution $\mathbb{P}(\varphi'|\varphi)$ need not be over *all* possible things that β might do, but could be over ontological categories that summarise β's possible actions.

In the absence of in-coming utterances, the conditional probabilities, $\mathbb{P}^t(\varphi'|\varphi)$, tend to ignorance as represented by a *decay limit distribution* $\mathbb{D}(\varphi'|\varphi)$. α may have background knowledge concerning $\mathbb{D}(\varphi'|\varphi)$ as $t \to \infty$, otherwise α may assume that it has maximum entropy whilst being consistent with the data. In general, given a distribution, $\mathbb{P}^t(X_i)$, and a decay limit distribution $\mathbb{D}(X_i)$, $\mathbb{P}^t(X_i)$ decays by:

$$\mathbb{P}^{t+1}(X_i) = \Gamma_i(\mathbb{D}(X_i), \mathbb{P}^t(X_i)) \tag{1}$$

where Γ_i is the *decay function* for the X_i satisfying the property that $\lim_{t\to\infty} \mathbb{P}^t(X_i) = \mathbb{D}(X_i)$. For example, Γ_i could be linear: $\mathbb{P}^{t+1}(X_i) = (1-\varepsilon_i) \times \mathbb{D}(X_i) + \varepsilon_i \times \mathbb{P}^t(X_i)$, where $\varepsilon_i < 1$ is the decay rate for the i'th distribution. Either the decay function or the decay limit distribution could also be a function of time: Γ_i^t and $\mathbb{D}^t(X_i)$.

If α receives an utterance, μ, from β then: if α did not know μ already and had some way of accommodating μ then we would expect the integrity of $\mathcal{M}^t$ to increase. Suppose that α receives a message μ from agent β at time t. Suppose that this message states that something is so with probability z, and suppose that α attaches an epistemic belief $\mathbb{R}^t(\alpha,\beta,\mu)$ to μ — this probability reflects α's level of personal *caution* — a method for estimating $\mathbb{R}^t(\alpha,\beta,\mu)$ is given in Section 3.2. Each of α's active plans, s, contains constructors for a set of distributions in the world model $\{X_i\} \in \mathcal{M}^t$ together with associated *update functions*, $J_s(\cdot)$, such that $J_s^{X_i}(\mu)$ is a set of linear constraints on the posterior distribution for X_i. These update functions are the link between the communication language and the internal representation. Denote the prior distribution $\mathbb{P}^t(X_i)$ by p, and let $p_{(\mu)}$ be the distribution with minimum relative entropy[2] with respect to p: $p_{(\mu)} = \arg\min_r \sum_j r_j \log \frac{r_j}{p_j}$ that satisfies the constraints $J_s^{X_i}(\mu)$. Then let $q_{(\mu)}$ be the distribution:

$$q_{(\mu)} = \mathbb{R}^t(\alpha,\beta,\mu) \times p_{(\mu)} + (1 - \mathbb{R}^t(\alpha,\beta,\mu)) \times p \tag{2}$$

and to prevent uncertain observations from weakening the estimate let:

$$\mathbb{P}^t(X_{i(\mu)}) = \begin{cases} q_{(\mu)} & \text{if } q_{(\mu)} \text{ is more interesting than } p \\ p & \text{otherwise} \end{cases} \tag{3}$$

[2] Given a probability distribution q, the *minimum relative entropy distribution* $p = (p_1, \ldots, p_I)$ subject to a set of J linear constraints $g = \{g_j(p) = a_j \cdot p - c_j = 0\}, j = 1, \ldots, J$ (that must include the constraint $\sum_i p_i - 1 = 0$) is: $p = \arg\min_r \sum_j r_j \log \frac{r_j}{q_j}$. This may be calculated by introducing Lagrange multipliers λ: $L(p,\lambda) = \sum_j p_j \log \frac{p_j}{q_j} + \lambda \cdot g$. Minimising L, $\{\frac{\partial L}{\partial \lambda_j} = g_j(p) = 0\}, j = 1, \ldots, J$ is the set of given constraints g, and a solution to $\frac{\partial L}{\partial p_i} = 0, i = 1, \ldots, I$ leads eventually to p. Entropy-based inference is a form of Bayesian inference that is convenient when the data is sparse [14] and encapsulates common-sense reasoning [15].

A general measure of whether $q_{(\mu)}$ is more interesting than p is: $\mathbb{K}(q_{(\mu)} \| \mathbb{D}(X_i)) > \mathbb{K}(p \| \mathbb{D}(X_i))$, where $\mathbb{K}(x \| y) = \sum_j x_j \ln \frac{x_j}{y_j}$ is the Kullback-Leibler distance between two probability distributions x and y.

Finally merging Eqn. 3 and Eqn. 1 we obtain the method for updating a distribution X_i on receipt of a message μ:

$$\mathbb{P}^{t+1}(X_i) = \Gamma_i(\mathbb{D}(X_i), \mathbb{P}^t(X_{i(\mu)})) \tag{4}$$

This procedure deals with integrity decay, and with two probabilities: first, the probability z in the percept μ, and second the belief $\mathbb{R}^t(\alpha, \beta, \mu)$ that α attached to μ.

The interaction between agents α and β will involve β making contractual commitments and (perhaps implicitly) committing to the truth of information exchanged. No matter what these commitments are, α will be interested in any variation between β's commitment, φ, and what is actually observed (as advised by the institution agent ξ), as the enactment, φ'. We denote the relationship between commitment and enactment, $\mathbb{P}^t(\text{Observe}(\varphi') | \text{Commit}(\varphi))$ simply as $\mathbb{P}^t(\varphi' | \varphi) \in \mathcal{M}^t$.

In the absence of in-coming messages the conditional probabilities, $\mathbb{P}^t(\varphi' | \varphi)$, should tend to ignorance as represented by the *decay limit distribution* and Eqn. 1. We now show how Eqn. 4 may be used to revise $\mathbb{P}^t(\varphi' | \varphi)$ as observations are made. Let the set of possible enactments be $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_m\}$ with prior distribution $p = \mathbb{P}^t(\varphi' | \varphi)$. Suppose that message μ is received, we estimate the posterior $p_{(\mu)} = (p_{(\mu)i})_{i=1}^m = \mathbb{P}^{t+1}(\varphi' | \varphi)$.

First, if $\mu = (\varphi_k, \varphi)$ is observed then α may use this observation to estimate $p_{(\varphi_k)k}$ as some value d at time $t+1$. We estimate the distribution $p_{(\varphi_k)}$ by applying the principle of minimum relative entropy as in Eqn. 4 with prior p, and the posterior $p_{(\varphi_k)} = (p_{(\varphi_k)j})_{j=1}^m$ satisfying the single constraint: $J^{(\varphi' | \varphi)}(\varphi_k) = \{p_{(\varphi_k)k} = d\}$.

Second, we consider the effect that the enactment ϕ' of another commitment ϕ, also by agent β, has on $p = \mathbb{P}^t(\varphi' | \varphi)$. Given the observation $\mu = (\phi', \phi)$, define the vector t as a linear function of semantic distance by:

$$t_i = \mathbb{P}^t(\varphi_i | \varphi) + (1 - \mid \delta(\phi', \phi) - \delta(\varphi_i, \varphi) \mid) \cdot \delta(\varphi', \phi)$$

for $i = 1, \dots, m$. t is not a probability distribution. The multiplying factor $\delta(\varphi', \phi)$ limits the variation of probability to those formulae whose ontological context is not too far away from the observation. The posterior $p_{(\phi', \phi)}$ is defined to be the normalisation of t.

3.2 Estimating Reliability

$\mathbb{R}^t(\alpha, \beta, \mu)$ is an epistemic probability that takes account of α's personal caution. An empirical estimate of $\mathbb{R}^t(\alpha, \beta, \mu)$ may be obtained by measuring the 'difference' between commitment and enactment. Suppose that μ is received from agent β at time u and is verified by ξ as μ' at some later time t. Denote the prior $\mathbb{P}^u(X_i)$ by p. Let $p_{(\mu)}$ be the posterior minimum relative entropy distribution subject to the constraints $J_s^{X_i}(\mu)$, and let $p_{(\mu')}$ be that distribution subject to $J_s^{X_i}(\mu')$. We now estimate what $\mathbb{R}^u(\alpha, \beta, \mu)$ should have been in the light of knowing *now*, at time t, that μ should have been μ'.

The idea of Eqn. 2, is that $\mathbb{R}^t(\alpha, \beta, \mu)$ should be such that, *on average* across $\mathcal{M}^t$, $q_{(\mu)}$ will predict $p_{(\mu')}$ — no matter whether or not μ was used to update the distribution for X_i, as determined by the condition in Eqn. 3 at time u. The *observed belief* in μ and distribution X_i, $\mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu'$, on the basis of the verification of μ with μ', is the value of k that minimises the Kullback-Leibler distance:

$$\mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu' = \arg\min_k \mathbb{K}(k\cdot p_{(\mu)} + (1-k)\cdot p \parallel p_{(\mu')})$$

The predicted *information* in the enactment of μ with respect to X_i is:

$$\mathbb{I}^t_{X_i}(\alpha,\beta,\mu) = \mathbb{H}^t(X_i) - \mathbb{H}^t(X_{i(\mu)}) \tag{5}$$

that is the reduction in uncertainty in X_i where $\mathbb{H}(\cdot)$ is Shannon entropy. Eqn. 5 takes account of the value of $\mathbb{R}^t(\alpha,\beta,\mu)$.

If $\mathbf{X}(\mu)$ is the set of distributions that μ affects, then the *observed belief* in β's promises on the basis of the verification of μ with μ' is:

$$\mathbb{R}^t(\alpha,\beta,\mu)|\mu' = \frac{1}{|\mathbf{X}(\mu)|}\sum_i \mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu' \tag{6}$$

If $\mathbf{X}(\mu)$ are independent the predicted *information* in μ is:

$$\mathbb{I}^t(\alpha,\beta,\mu) = \sum_{X_i\in\mathbf{X}(\mu)} \mathbb{I}^t_{X_i}(\alpha,\beta,\mu) \tag{7}$$

Suppose α sends message μ to β where μ is α's private information, then assuming that β's reasoning apparatus mirrors α's, α can estimate $\mathbb{I}^t(\beta,\alpha,\mu)$. For each formula φ at time t when μ has been verified with μ', the *observed belief* that α has for agent β's promise φ is:

$$\mathbb{R}^{t+1}(\alpha,\beta,\varphi) = (1-\chi)\times\mathbb{R}^t(\alpha,\beta,\varphi) + \chi\times\mathbb{R}^t(\alpha,\beta,\mu)|\mu'\times\delta(\varphi,\mu)$$

where δ measures the semantic distance between two sections of the ontology as introduced in Section 2, and χ is the learning rate. Over time, α notes the context of the various μ received from β, and over the various combinations of utterance category, and position in the ontology, and aggregates the belief estimates accordingly. For example: "I believe John when he promises to deliver good cheese, but not when he is discussing the identity of his wine suppliers."

3.3 Measuring Accumulated Evidence

α's world model, $\mathcal{M}^t$, is a set of probability distributions. If at time t, α receives an utterance u that may alter this world model (as described in Section 3.1) then the (Shannon) *information* in u with respect to the distributions in $\mathcal{M}^t$ is: $\mathbb{I}(u) = \mathbb{H}(\mathcal{M}^t) - \mathbb{H}(\mathcal{M}^{t+1})$. Let $\mathcal{N}^t \subseteq \mathcal{M}^t$ be α's model of agent β. If β sends the utterance u to α then the *information* about β within u is: $\mathbb{H}(\mathcal{N}^t) - \mathbb{H}(\mathcal{N}^{t+1})$. We note that by defining information in terms of the change in uncertainty in $\mathcal{M}^t$ our measure is based on the way in which that update is performed that includes an estimate of the 'novelty' or 'interestingness' of utterances in Eqn 3.

3.4 Building the Map

We give structure to the measurement of accumulated evidence using an *illocutionary framework* to categorise utterances, and an *ontology*. The illocutionary framework will depend on the nature of the interactions between the agents. The LOGIC framework for argumentative negotiation [16] is based on five categories: Legitimacy of the arguments, Options i.e. deals that are acceptable, Goals i.e. motivation for the negotiation, Independence i.e: outside options, and Commitments that the agent has including its assets. The LOGIC framework contains two models: first α's model of β's private information, and second, α's model of the private information that β has about α. Generally we assume that α has an illocutionary framework $\mathcal{F}$ and a categorising function $v : U \rightarrow \mathcal{P}(\mathcal{F})$ where U is the set of utterances. The power set, $\mathcal{P}(\mathcal{F})$, is required as some utterances belong to multiple categories. For example, in the LOGIC framework the utterance "I will not pay more for apples than the price that John charges" is categorised as both Option and Independence.

In [16] two central concepts are used to describe relationships and dialogues between a pair of agents. These are *intimacy* — degree of closeness, and *balance* — degree of fairness. Both of these concepts are summary measures of relationships and dialogues, and are expressed in the LOGIC framework as 5×2 matrices. A different and more general approach is now described. The intimacy of α's relationship with β_i, I^t_i, measures the amount that α knows about β_i's private information and is represented as real numeric values over $\mathcal{G} = \mathcal{F} \times \mathcal{O}$. Suppose α receives utterance u from β_i and that category $f \in v(u)$. For any concept $c \in \mathcal{O}$, define $\Delta(u,c) = \max_{c' \in u} \delta(c',c)$. Denote the value of I^t_i in position (f,c) by $I^t_{i(f,c)}$ then: $I^t_{i(f,c)} = \rho \times I^{t-1}_{i(f,c)} + (1-\rho) \times \mathbb{I}(u) \times \Delta(u,c)$ for any c, where ρ is the discount rate. The *balance* of α's relationship with β_i, B^t_i, is the element by element numeric difference of I^t_i and α's estimate of β_i's intimacy on α.

4 Not Doing the 'Wrong Thing'

We now describe our second 'map' of the trust that represents our agent's accumulated, time-discounted belief that the observed agent will act in a way that fails to respect the confidentiality of previously passed information. Having built much of the machinery above, the description of the second map is simpler than the first.

[16] advocates the controlled revelation of information as a way of managing the intensity of relationships. Information that becomes public knowledge is worthless, and so respect of confidentiality is significant to maintaining the value of revealed private information. We have not yet described how to measure the extent to which one agent respects the confidentiality of another agent's information — that is, the strength of belief that another agent will respect the confidentially of my information: both by not passing it on, and by not using it so as to disadvantage me.

Consider the motivating example, α sells a case of apples to β at cost, and asks β to treat the deal in confidence. Moments later another agent β' asks α to quote on a case of apples — α might then reasonably increase his belief in the proposition that β had spoken to β'. Suppose further that α quotes β' a fair market price for the apples and that β' rejects the offer — α may decide to further increase this belief. Moments later β

offers to purchase another case of apples for the same cost. α may then believe that β may have struck a deal with β' over the possibility of a cheap case of apples.

This aspect of trust is the mirror image of trust that is built by an agent "doing the right thing" — here we measure the extent to which an agent does *not* do the wrong thing. As human experience shows, validating respect for confidentiality is a tricky business. In a sense this is the 'dark side' of trust. One proactive ploy is to start a false rumour and to observe how it spreads. The following reactive approach builds on the apples example above.

An agent will know when it passes confidential information to another, and it is reasonable to assume that the significance of the act of passing it on decreases in time. In this simple model we do not attempt to value the information passed as in Section 3.3. We simply note the amount of confidential information passed and observe any indications of a breach of confidence.

If α sends utterance u to β "in confidence", then u is categorised as f as described in Section 3.4. C_i^t measures the amount of confidential information that α passes to β_i in a similar way to the intimacy measure I_i^t described in Section 3.4: $C_{i(f,c)}^t = \rho \times C_{i(f,c)}^{t-1} + (1-\rho) \times \Delta(u,c)$, for any c where ρ is the discount rate; if no information is passed at time t then: $C_{i(f,c)}^t = \rho \times C_{i(f,c)}^{t-1}$. C_i^t represents the time-discounted amount of confidential information passed in the various categories.

α constructs a companion framework to C_i^t, L_i^t is as estimate of the amount of information leaked by β_i represented in $\mathcal{G}$. Having confided u in β_i, α designs update functions J_u^L for the L_i^t as described in Section 3.1. In the absence of evidence imported by the J_u^L functions, each value in L_i^t decays by: $L_{i(f,c)}^t = \xi \times L_{i(f,c)}^{t-1}$, where ξ is in $[0,1]$ and probably close to 1. The J_u^L functions scan every observable utterance, u', from each agent β' for evidence of leaking the information u, $J_u^L(u') = \mathbb{P}(\beta' \text{ knows } u \mid u' \text{ is observed})$. As previously: $L_{i(f,c)}^t = \xi \times L_{i(f,c)}^{t-1} + (1-\xi) \times J_u^L(u') \times \Delta(u,c)$ for any c.

This simple model estimates C_i^t the amount of confidential information passed, and L_i^t the amount of presumed leaked, confidential information represented over $\mathcal{G}$. The 'magic' is in the specification of the J_u^L functions. A more exotic model would estimate "who trusts who more than who with what information" — this is what we have elsewhere referred to as a *trust network* [17]. The feasibility of modelling a trust network depends substantially on how much detail each agent can observe in the interactions between other agents.

5 Summary Measures

[17] describes measures of: *trust* (in the execution of contracts), *honour* (validity of argumentation), and *reliability* (of information). The execution of contracts, soundness of argumentation and correctness of information are all represented as conditional probabilities $\mathbb{P}(\varphi'|\varphi)$ where φ is an expectation of what may occur, and φ' is the subsequent observation of what does occur.

These summary measures are all abstracted using the ontology; for example, "What is my trust of John for the supply of red wine?". These measures are also used to summarise the information in some of the categories in the illocutionary framework. For example, if these measures are used to summarise estimates $\mathbb{P}^t(\varphi'|\varphi)$ where φ is a deep motivation of β's (i.e. a Goal), or a summary of β's financial situation (i.e. a Commitment) then this contributes to a sense of trust at a deep social level.

The measures here generalise what are commonly called *trust*, *reliability* and *reputation* measures into a single computational framework. It they are applied to the execution of contracts they become trust measures, to the validation of information they become reliability measures, and to socially transmitted overall behaviour they become reputation measures.

Ideal enactments. Consider a distribution of enactments that represent α's "ideal" in the sense that it is the best that α could reasonably expect to happen. This distribution will be a function of α's *context* with β denoted by e, and is $\mathbb{P}^t_I(\varphi'|\varphi,e)$. Here we use relative entropy to measure the difference between this ideal distribution, $\mathbb{P}^t_I(\varphi'|\varphi,e)$, and the distribution of expected enactments, $\mathbb{P}^t(\varphi'|\varphi)$. That is:

$$M(\alpha,\beta,\varphi) = 1 - \sum_{\varphi'} \mathbb{P}^t_I(\varphi'|\varphi,e)\log\frac{\mathbb{P}^t_I(\varphi'|\varphi,e)}{\mathbb{P}^t(\varphi'|\varphi)} \tag{8}$$

where the "1" is an arbitrarily chosen constant being the maximum value that this measure may have.

Preferred enactments. Here we measure the extent to which the enactment φ' is preferable to the commitment φ. Given a predicate Prefer(c_1,c_2,e) meaning that α prefers c_1 to c_2 in environment e. An evaluation of $\mathbb{P}^t(\text{Prefer}(c_1,c_2,e))$ may be defined using $\delta(\cdot)$ and the evaluation function $w(\cdot)$ — but we do not detail it here. Then if $\varphi \leq o$:

$$M(\alpha,\beta,\varphi) = \sum_{\varphi'} \mathbb{P}^t(\text{Prefer}(\varphi',\varphi,o))\mathbb{P}^t(\varphi' \mid \varphi)$$

Certainty in enactment. Here we measure the consistency in expected acceptable enactment of commitments, or "the lack of expected uncertainty in those possible enactments that are better than the commitment as specified". If $\varphi \leq o$ let: $\Phi_+(\varphi,o,\kappa) = \{\varphi' \mid \mathbb{P}^t(\text{Prefer}(\varphi',\varphi,o)) > \kappa\}$ for some constant κ, and:

$$M(\alpha,\beta,\varphi) = 1 + \frac{1}{B^*}\cdot \sum_{\varphi'\in\Phi_+(\varphi,o,\kappa)} \mathbb{P}^t_+(\varphi'|\varphi)\log\mathbb{P}^t_+(\varphi'|\varphi)$$

where $\mathbb{P}^t_+(\varphi'|\varphi)$ is the normalisation of $\mathbb{P}^t(\varphi'|\varphi)$ for $\varphi' \in \Phi_+(\varphi,o,\kappa)$,

$$B^* = \begin{cases} 1 & \text{if } |\Phi_+(\varphi,o,\kappa)| = 1 \\ \log|\Phi_+(\varphi,o,\kappa)| & \text{otherwise} \end{cases}$$

6 Conclusion

Trust is evaluated by applying summary measures to a rich model of interaction that is encapsulated in two maps. The first map gives a fine-grained view of an agent's accumulated, time-discounted belief that the enactment of commitments by another agent will be in-line with what was promised. The second map contains estimates of the accumulated, time-discounted belief that the observed agent will act in a way that fails to respect the confidentiality of previously passed information. The structure of these maps is defined in terms of a categorisation of utterances and the ontology. Three summary measures are described that may be used to give a succinct view of trust.

References

1. Reece, S., Rogers, A., Roberts, S., Jennings, N.R.: Rumours and reputation: Evaluating multi-dimensional trust within a decentralised reputation system. In: 6th International Joint Conference on Autonomous Agents and Multi-agent Systems AAMAS 2007 (2007)
2. Ramchurn, S., Huynh, T., Jennings, N.: Trust in multi-agent systems. The Knowledge Engineering Review 19, 1–25 (2004)
3. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez, J.A., Sierra, C.: Environment engineering for multiagent systems. Journal on Engineering Applications of Artificial Intelligence 18 (2005)
4. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24, 33–60 (2005)
5. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 5, 58–71 (2007)
6. Viljanen, L.: Towards an Ontology of Trust. In: Katsikas, S.K., López, J., Pernul, G. (eds.) TrustBus 2005. LNCS, vol. 3592, pp. 175–184. Springer, Heidelberg (2005)
7. Huynh, T., Jennings, N., Shadbolt, N.: An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems 13, 119–154 (2006)
8. MacKay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
9. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. International Journal of Group Decision and Negotiation 10, 199–215 (2001)
10. Faratin, P., Sierra, C., Jennings, N.: Using similarity criteria to make issue trade-offs in automated negotiation. Journal of Artificial Intelligence 142, 205–237 (2003)
11. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter. The MIT Press, Cambridge (1994)
12. Kraus, S.: Negotiation and cooperation in multi-agent environments. Artificial Intelligence 94, 79–97 (1997)
13. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15, 871–882 (2003)
14. Cheeseman, P., Stutz, J.: On The Relationship between Bayesian and Maximum Entropy Inference. In: Bayesian Inference and Maximum Entropy Methods in Science and Engineering, pp. 445–461. American Institute of Physics, Melville (2004)
15. Paris, J.: Common sense and maximum entropy. Synthese 117, 75–93 (1999)
16. Sierra, C., Debenham, J.: The LOGIC Negotiation Model. In: Proceedings Sixth International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2007, Honolulu, Hawai'i (2007)
17. Sierra, C., Debenham, J.: Trust and honour in information-based agency. In: Stone, P., Weiss, G. (eds.) Proceedings Fifth International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2006, Hakodate, Japan, pp. 1225–1232. ACM Press, New York (2006)

Implementation of a TCG-Based Trusted Computing in Mobile Device[*]

SuGil Choi, JinHee Han, JeongWoo Lee, JongPil Kim, and SungIk Jun

Wireless Security Application Research Team
Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, South Korea
{sooguri,hanjh,jeow7,kimjp,sijun}@etri.re.kr

Abstract. Our implementation is aimed at estimating the possibility of employing TCG-based trusted computing mechanisms, such as verifying the code-integrity of executables and libraries at load-time and remote attestation, in mobile devices. Considering the restrained resource in mobile device, the experimentation shows promising results, thereby enabling these mechanisms to be used as a basic building block for a more secured mobile service. To this end, we add a new feature of integrity measurement and verification to Wombat Linux kernel and Iguana embedded OS. We also implement attestation agents, Privacy CA, and TCG Software Stack.

1 Introduction

The wide use and increasing capabilities of mobile devices introduce security risks to the mobile phone users as well as mobile operators. Mobile viruses will become a costly problem for many operators that cause subscriber dissatisfaction. Virus writers are attempting to disrupt mobile networks through infected MMS messages or harm mobile devices with viruses. There is evidence that virus writers are re-focusing their energy from the PC world to the still widely unprotected mobile environment. These security breaches are something anyone wants to avoid and the technology for preventing them has been developed, such as antivirus and firewall against mobile threats, and USIM (Universal Subscriber Identity Module).

However, the defense measures of antivirus and firewall have been proved not to be enough to secure computer system and this conclusion will be also applied in mobile environment. USIM is employed in the wireless cellular networks to authenticate users, but it can't guarantee that the mobile device is trustworthy. One of the security challenges to make up for the the weak points as shown above is provisioning of building blocks for trusted computing. Trusted Computing can provide the following properties within the mobile context, which are useful for a range of services [6].

[*] This work was supported by the IT R&D program of MIC/IITA. [2006-S-041-02, Development of a common security core module for supporting secure and trusted service in the next generation mobile terminals].

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 18–27, 2008.

- enabling a user to have more confidence in the behavior of their mobile platform. In particular, users can have more trust in their platform to handle private data.
- recognizing that a platform has known properties. This is useful in situations such as allowing mobile platform to access a corporate network and providing remote access via a known public access point

The Trusted Computing Group (TCG) specification [1] aims to address this problem and Mobile Phone Work Group (MPWG) in TCG specifically deals with trusted computing in mobile environment. The specifications defined by TCG describe functionalities to address the aforementioned issues. First, the method of securing a computing platform in a trusted state is called Integrity Measurement and Verification (IMV). Second, the process of proving its state to remote entity is called attestation. The implementation of this concept in PC environment appears in [5], but our system is the first to extend the TCG-based concepts to mobile environment.

We modify Wombat Linux kernel in order to measure and verify the integrity of binary executables and libraries as soon as they are loaded. Wombat is a NICTA's architecture-independent para-virtualised Linux for L4-embedded MicroKernel [11] and we will see mobile phones with Wombat Linux on top of L4. In order to verify the integrity of the code executed, Reference Integrity Metric (RIM) certificate called RIM_Cert is used, which is a structure authorizing a measurement value that is extended into a Platform Configuration Register (PCR) defined in the RIM_Cert. RIM_Cert is a new feature introduced in MPWG [2][3]. We wrote a program called RIMCertTool for generating a RIM_Cert which is inserted into a section in Executable and Linkable Format (ELF) file. As, nowadays, ELF is the standard format for Linux executables and libraries, we use only ELF file for our executables and libraries. In this way, RIM_Cert can be delivered to mobile device without any additional acquisition mechanism. To prove to a remote party what codes were executed, mobile devices need to be equipped with TCG Software Stack (TSS) [4] and Mobile Trusted Module (MTM) [3], and Certification Authority called Privacy CA should be working. We implement most of these components and set up a system. As the mobile devices are resource-constrained compared to PC, security features such as IMV and attestation should come with little overhead. Our experimental results show a very small overhead at load time of executables and libraries in mobile device. Further, it is likely that uninterrupted mobile service preceded with attestation is feasible.

The rest of the paper is organized as follows. Next, we give some overview on TCG specification focusing on IMV and attestation. In Section 3, we describe the implementation of our approach. Section 4 describes the experiments that highlight the performance impact by our system. Section 5 sketches enhancements to our system that are being planned and is followed by a conclusion in Section 6.

2 Overview of TCG-Based Trusted Computing

TCG specification requires the addition of a cryptographic processor chip to the platform, called a Trusted Platform Module (TPM). The TPM must be a fixed part of the platform that cannot be removed from the platform and transferred to another platform. The TPM provides a range of cryptographic primitives including SHA-1 hash, and signing and verification using RSA. There are also protected registers called PCR. MPWG defines a new specification on MTM which adds new commands and structures to existing TPM specification in order to enable trusted computing in a mobile device context.

Integrity Measurement and Verification (IMV): A measurement is done by hashing the binary image of entities, such as OS and executables, with SHA-1. A measurement result is stored by extending a particular PCR as follows. A new measurement value is concatenated with the current PCR value and then hashed by SHA-1. The result is stored as a new value of the PCR. The extend operation works like this: (where | denotes concatenation)

ExtendedPCRValue = SHA1(Previous PCR Value | new measurement value)
In order to verify the measurement result, RIM values need to be available, and the authenticity and integrity of them should be preserved. These requirements are met by RIM_Certs [2]. The RIM is included in a RIM_Cert which is issued by a CA called RIM_Auth and MTM has a pre-configured public key of a Root CA. The public key of the Root CA is termed Root Verification Authority Identifier (RVAI). The Root CA can delegate the role of issuing RIM_Certs to RIM_Auths by issuing certificates called RIM_Auth_Certs or may directly sign the RIM_Certs. As the MTM is equipped with the RVAI, the verification of RIM_Cert takes place inside a MTM. Considering two entities A (Agent for IMV) and T (Target of IMV), the measurement and verification operation is as follows:

1. A measures T. The result is a T's hash value
2. A retrieves the RIM from RIM_Cert which is embedded in T's ELF file and checks if the T's hash value matches the RIM
3. If those matches, A requests the verification of the RIM_Cert to the MTM
4. If the verification of the RIM_Cert is successful, the MTM extends the RIM into a PCR
5. T's hash value and its related information (e.g file name, extended PCR index) are stored in a Measurement Log (ML) which resides in a storage outside a MTM
6. The execution of T is allowed

Remote Attestation: Simple description of attestation protocol used by the challenger (C) to securely validate integrity claims of the remote platform (RP) is as follows:

1. C : generates random number (nonce) of 20 bytes
2. C −> RP : nonce
3. RP : load AIK_{priv} into MTM
4. RP : retrieve $Quote = sig(PCRs, nonce)_{AIK_{priv}}$, PCRs, nonce
5. RP : retrieve Measurement Log (ML)
6. RP −> C : $Quote$, ML, $Cert(AIK_{pub})$

7. C : verify $Cert(AIK_{pub})$
8. C : validate $sig(PCRs, nonce)_{AIK_{priv}}$ using AIK_{pub}
9. C : validate nonce and ML using PCRs

The AIK is created securely inside the MTM and the corresponding public key AIK_{pub} can be certified by a trusted party called Privacy CA. There should be an attestation agent at the RP which interacts with the Privacy CA to create a $Cert(AIK_{pub})$, waits attestation request, prepares response message, and sends it to the challenger. In step 4, the attestation agent sends a *Quote* request to the MTM by calling a relevant function in TSS and the MTM signs the current PCR values together with the given nonce using AIK_{priv}. In step 7, challenger determines if the $Cert(AIK_{pub})$ is trusted. In step 8, the successful verification of the *Quote* with AIK_{pub} shows that the RP has a correct configuration with trusted MTM, but the challenger can't get any information to identify the device. In step 9, tampering with the ML is made visible by walking through the ML and re-computing the PCRs (simulating the PCR extend operations as described in the previous subsection) and comparing the result with the PCRs included in the *Quote* received. If the re-computed PCRs match the signed PCRs, then the ML is valid. For further detail, please refer to [5].

3 Implementation

In this section, we discuss how we realize the concept of IMV and attestation described in Section 2. We first describe how only the trusted executables and libraries can be loaded into memory on a mobile device that runs Wombat Linux on top of L4 MicroKernel and has a MTM emulation board attached to it. Then we explain the components implemented to support attestation mechanism.

Fig 1 shows the system configuration and the explanation about this will be given in the relevant parts following.

We port L4-embedded MicroKernel, Iguana embedded OS, and Wombat Linux onto a mobile device which is used for viewing Digital Media Broadcasting. L4-embedded is a promising MicroKernel as its deployment on the latest Qualcomm CDMA chipsets shows, thus we decide to employ it. As the work on making MTM chip is going on, MTM emulation board is connected to the DMB device. The other researchers in our team are making an effort to make a MTM chip and MTM emulation board is the product in an experimental stage. It supports hardware cryptographic engine, command processing, and etc. The detailed explanation on this will be given in another paper by the developers.

We make enhancements to the Wombat Linux and Iguana to implement the measurement and verification functionalities. We insert a measurement function call into where executables and libraries are loaded, specifically do_mmap_pgoff in mmap.c. The steps after calling measurement function are depicted in Fig 2.

The measurement function takes file struct as argument, and file name and the content of the file can be accessed using the file struct. For the inclusion of RIM_Cert, we introduce a new type of ELF section called RIM_Cert section. We

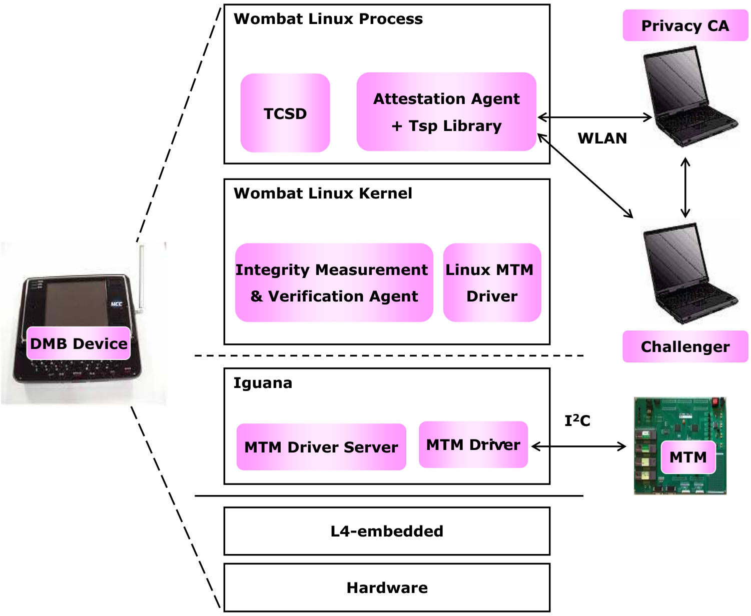


Fig. 1. System Configuration

created a tool for generating RIM_Cert which embeds an RSA signature of all text and data segments. A RIM_Cert consists of a set of standard information and a proprietary authentication field which include PCR index for extension, expected measurement value, integrity check data, and key ID for integrity verification.

The process of measurement is obtaining a Target Integrity Metric (TIM) by hashing text and data segments. In order to increase the performance of verification, two kinds of cache are employed. One is Whist List (WL) for recording the TIMs of trusted files and another is Black List (BL) for untrusted files. If the verification is correct, then the TIM is cached in WL and, in subsequent loads, the verification steps can be skipped only if the TIM from measurement is found in WL. If the TIM is found in BL, the execution of corresponding binary or library is prevented. The conditions for verification success are as follows: TIM matches RIM and RIM_Cert is trusted. The process of checking if the RIM_Cert was signed by trusted party takes place inside the MTM.

We implement a MTM Driver for communicating with MTM board through I^{2}C bus, MTM Driver server for passing data between MTM Driver and Linux MTM Driver, and Linux MTM Driver as shown in Fig 1. The Linux MTM Driver connects to MTM Driver server via L4 IPC. We implement a function RIMCert_Verify_Extend() in the Linux MTM Driver which takes RIM_Cert as argument and returns the verification result of the RIM_Cert from the MTM board. We also implement a function PCR_Extend() which takes a PCR index and TIM and then returns the extended value from the MTM board. For simplicity, the Root CA directly signs the RIM_Cert and the RVAI which is the public

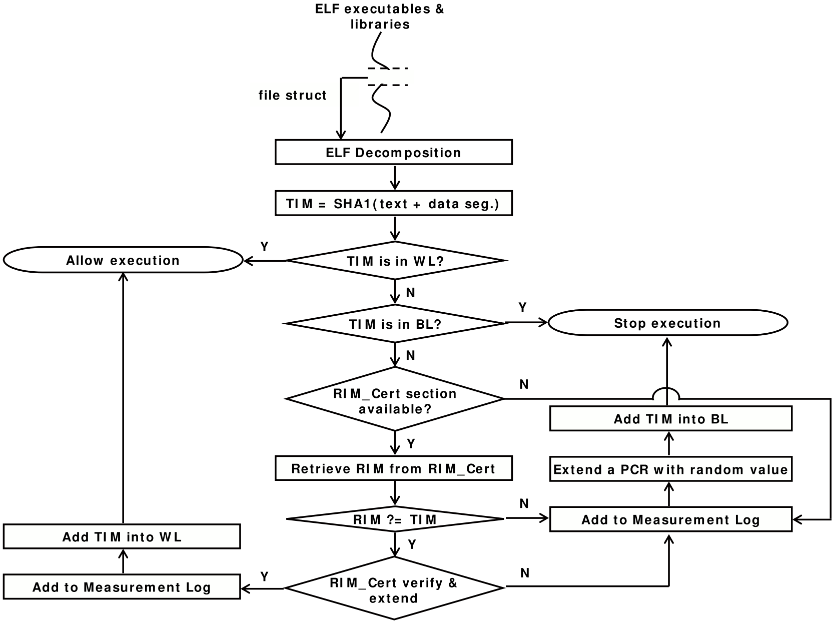


Fig. 2. Sequence of Integrity Measurement and Verification

key of the Root CA is stored inside the MTM board. The Measurement Log is recorded using Proc file system which is efficient by writing to and reading from memory. We implement a Read function for the ML Proc file, thus attestation agent in user-level can read the ML.

We implement a TCG Core Service Daemon (TCSD) and TCG Service Provider (TSP) library in order to support remote attestation. Attestation agent calls relevant functions in TSP library which connects to TCSD. The TCSD takes the role of passing data to and from MTM board through Linux MTM Driver. DMB device communicates with Privacy CA and Challenger through Wireless LAN. We use commercial ASN.1 compiler to create and validate AIK Certificates. The Privacy CA and Challenger are Linux application running on laptop computers.

4 Experimental Results

Experimental results show that load-time integrity check of executables and libraries can be performed with reasonable performances and thus our design and implementation are a practical mechanism. Table 1 shows the results of performance measurement of running some of the executables and libraries on DMB device. The device consists of a 520 MHz PXA270 processor with 128 MB memory running Wombat Linux on top of L4-embedded MicroKernel. The MTM emulation board has the following features: 19.2 MHz EISC3280H microprocessor, 16 KB memory, 32 KB EEPROM for data storage, and 400 kbps I^{2}C Interface.

Table 1. Performance of Measurement and Verification (in sec)

Target	**Size**	**Measurement**	**Verification**	**RIM_Cert**
ld-linux.so	115668, 72004	0.02	0.41	0.40
libm.so	780380, 442164	0.1	0.41	0.40
libssl.so	217324, 126848	0.03	0.42	0.40
libnss_files.so	54324, 28032	0.00	0.42	0.40
wlanconfig	74720, 45304	0.01	0.43	0.40
tcsd	314032, 111756	0.04	0.41	0.40
attestation agent	76440, 41632	0.01	0.42	0.40
busybox	358536, 276312	0.05	0.43	0.40

The first number in the field of **Size** is the entire file size and the second one is the size of text and data segments. **RIM_Cert** field represents the time taken from sending request of RIM_Cert Verification and Extend to the MTM board to getting the response from it. Each figure is the average time of running a test 10 times using the do_gettimeofday() function supported in Linux Kernel. The delay by verification is almost static and most of the delay comes from RIM_Cert Verification and Extend. The signature verification with RSA 2048 public key and relatively slow data transmission speed of I^2C may contribute to the delay. The overhead due to measurement grows with the size of text and data segments as the input for hash operation increases. Our experiment shows that the initial loading of executables or libraries can be delayed up to 0.51 sec and this overhead decreases to less than or equal to 0.1 sec with the introduction of cache, since no verification is required. libm.so is one of the large libraries in embedded Linux and the loading of it with cache takes 0.1 sec thus we believe that the overhead shown is not critical.

We also perform an experimentation to assess the time taken for attestation. The system for running Privacy CA and Challenger is the IBM ThinkPad notebook which uses an Intel CPU running at 2 GHz and has 2 GB RAM. Privacy CA and Challenger communicate with DMB device in the same subnet over wireless LAN. Table 2 summarizes the performance of attestation with 4 measurement entries in ML and 14 measurement entries in ML. Each tests is conducted 10 times and the result is the average of them. The meaning of each fields is as follows: **Attestation**: the entire time taken for attestation, **Client**: preparing attestation response message at mobile device, **Quote**: retrieving a Quote message from a MTM, **OIAP**: creation of authorization session with a MTM using Object-Independent Authorization Protocol (OIAP) , **Challenge**:

Table 2. Performance of Attestation (in sec)

	Attestation	**Client**	**Quote**	**OIAP**	**Challenge**	**Verification**
4 entries	1.9416	1.63	0.856	0.36	0.0461	0.006
14 entries	1.9908	1.63	0.856	0.36	0.0497	0.0062

preparing attestation challenge message, and **Verification**: verifying the attestation response message.

As shown above, the attestation can be completed within 2 seconds and this is believed to be a promising result considering further optimization of TSS and MTM. The creation of attestation response message at DMB device takes about 83 percent of the time taken for attestation. The retrieval of Quote message from the MTM and creation of OIAP session with the MTM, respectively, contribute 53 percent and 22 percent of the time for attestation response creation. These two tests are done by measuring how long it takes to return back after calling Linux MTM Driver. Thus, data transmission to and from a MTM and processing inside a MTM take 75 percent of the time for attestation response creation. The Quote operation is the most expensive and this is understandable because the operation requires interaction with the MTM through I^2C bus and includes signing with RSA 2048 private key. As the number of measurement entries increases, attestation takes a little longer but the difference is not significant. The difference may be attributed to the longer delay to transfer it over wireless LAN. The number of measurement entries increases by 10, which means SHA-1 operation should be conducted 10 times more at Challenger, but the overhead due to this is negligible because the time for verification grows just by 0.2 microseconds. The overhead for preparing attestation challenge is subject to large fluctuations as random number generation can be done in short time or take long. The generation of random number is done using RAND_bytes() function supported in openssl. The experimentation over other kinds of communication channel will produce different result, maybe longer delay because wireless LAN can deliver data at relatively high speed and all the participants in this experimentation are in the same subnet.

5 Future Work

Our implementation makes it possible to determine the code-integrity of executables and libraries at load-time, but it doesn't prevent modifying code existing in memory or executing injected code. Without the guarantee that loaded code is not vulnerable to attack during its operation, the decision about the platform integrity lacks confidence. Even worse is the fact that the kernel code in memory can be manipulated. According to the [10], there are at least three ways in which an attacker can inject code into a kernel as follows: loading a kernel module into a kernel, exploiting software vulnerabilities in the kernel code, and corrupting kernel memory via DMA writes. As the mechanism of integrity measurement and verification is realized as part of the kernel, the compromise of kernel can lead to the disruption of measurement and verification process. However, the measurement of the running kernel can't be easily represented with a hash as discussed in [9] and we also need to figure out where to place the functionality of measuring the running kernel. The fist issue is rather general problem as it is hard to measure running processes, either it is application or kernel. The latter issue can be solved by leveraging virtualization technology which provides separation between the measurement functionality and the measurement target.

As stated before, Wombat Linux kernel runs on top of L4-embedded MicroKernel and Iguana embedded OS which form a Virtual Machine Monitor (VMM). Iguana is a basic framework on which embedded systems can be built and provides services such as memory management, naming, and support for device drivers. Iguana consists of several threads with their own functionalities. As Iguana supports memory protection to provide isolation between guest OSes by encouraging a non-overlapping address-space layout and can keep track of allocated memory using objects called memsections, it is best to implement the agent for measuring the running kernel as a thread running along with other threads forming Iguana. L4 and Iguana form a Trusted Computing Base (TCB), thus the agent is always trusted to reliably measure Wombat Linux kernel during operation. We plan to create a new thread running as the measurement and verification agent, but how to measure the running kernel needs to be investigated further. In addition, the White List which resides in kernel-memory can also be corrupted, thus we need to monitor some security-critical memory regions.

We implement TSP and TCS following the specification [4] which is originally targeted for PC platforms. The implementation needs to be optimized considering the restrained resource in mobile device. The TSP and TCS communicate with each other over a socket connection, but this might not be a best solution for exchanging data between processes. Thus, we plan to find and implement more efficient way of inter-process data exchange. These enhancements will help reduce the time taken for attestation.

6 Conclusion

We have presented an implementation of TCG-based trusted computing in mobile environment and provided an analysis of experimental results. Central to our implementation is that it was realized on real mobile device running L4 MicroKernel which is one of the next-generation OS for mobile platforms and thus further improvements to verify the running kernel becomes viable. The experimental results are a proof that integrity measurement and verification specified in TCG can really work in mobile device without serious performance degradation. We hope this paper will motivate others in the field to embrace this technology, extend it, and apply it to build secure mobile systems.

References

1. Trusted Computing Group: TCG Specification Architecture Overview. Specification Revision 1.4, August 2 (2007), http://www.trustedcomputinggroup.org
2. Trusted Computing Group: TCG Mobile Reference Architecture. Specification version 1.0. Revision 1, June 12 (2007), http://www.trustedcomputinggroup.org
3. Trusted Computing Group: TCG Mobile Trusted Module Specification. Specification version 1.0. Revision 1, June 12 (2007), http://www.trustedcomputinggroup.org
4. Trusted Computing Group: TCG Software Stack. Specification version 1.2, March 7 (2007), http://www.trustedcomputinggroup.org

5. Sailer, R., Zhang, X., Jaeger, T., van Doorn, L.: Design and Implementation of a TCG-based Integrity Measurement Architecture. In: 13th Usenix Security Symposium (August 2004)
6. Pearson, S.: How trusted computers can enhance privacy preserving mobile applications. In: Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (June 2005)
7. Apvrille, A., Gordon, D., Hallyn, S., Pourzandi, M., Roy, V.: DigSig: Run-time Authentication of Binaries at Kernel Level. In: 18th Large Installation System Administration Conference, November 14 (2004)
8. van Doorn, L., Ballintijn, G.: Signed Executables for Linux. Tech. Rep. CS-TR-4259, University of Maryland, College Park, June 4 (2001)
9. Loscocco, P.A., Wilson, P.W., Pendergrass, J.A., McDonell, C.D.: Linux kernel integrity measurement using contextual inspection. In: ACM workshop on Scalable trusted computing, November 2 (2007)
10. Seshadri, A., Luk, M., Qu, N., Perrig, A.: SecVisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity OSes. In: ACM Symposium on Operating Systems Principles, October 14-17 (2007)
11. L4-embedded, http://www.ertos.nicta.com.au/research/l4/

A Model for Trust Metrics Analysis*

Isaac Agudo, Carmen Fernandez-Gago, and Javier Lopez

Department of Computer Science, University of Malaga, 29071, Málaga, Spain
{isaac,mcgago,jlm}@lcc.uma.es

Abstract. Trust is an important factor in any kind of network essential, for example, in the decision-making process. As important as the definition of trust is the way to compute it. In this paper we propose a model for defining trust based on graph theory and show examples of some simple operators and functions that will allow us to compute trust.

1 Introduction

In the recent years trust has become an important factor to be considered in any kind of social or computer network. The concept of trust in Computer Science derives from the concept on sociological or psychological environments. Trust becomes essential when an entity needs to establish how much trust to place on another of the entities in the system.

The definition of trust is not unique. It may vary depending on the context and the purpose where it is going to be used. For the approach adopted in this paper we will define trust as the *level of confidence* that an entity participating in a network system places on another entity of the same system for performing a given task. We mean by a *task* any action that an agent or entity in the system is entitled or in charged of performing.

Trust management systems have been introduced in order to create a coherent framework to deal with trust. The first attempts for developing trust management systems were PolicyMaker [5], KeyNote [4] or REFEREE [7]. Since the importance of building trust models has become vital for the development of some nowadays computer systems the way this trust is derived, i.e., the *metrics*, becomes also crucial. Metrics become very important for the deployment of these trust management systems as the way of quantifying trust. The simplest way to define a trust metric is by using a discrete model where an entity can be either 'trusted' or 'not trusted'. This can also be expressed by using numerical values such as 1 for trusted and 0 for not trusted. The range of discrete categories of trust can be extended with 'medium trust', 'very little trust' or 'a lot of trust', for example. More complex metrics use integer or real numbers, logical formulae like BAN logic [6] or vector like approaches [9]. In the early nineties the first proposals for trust metrics were developed in order to support Public Key Infrastructure (for instance [13]). In the recent years the development of new networks or systems such as P2P or

* This work has been partially funded by the European Commission through the research project SPIKE (FP7-ICT-2007-1-217098), and the Spanish Ministry of Science and Education through the research project ARES (CONSOLIDER CSD2007-00004).

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 28–37, 2008.

Ad-Hoc networks, or ubiquitous or mobile computing has led to the imminent growth of the development of trust management systems and consequently metrics for them. Most of the used metrics are based in probabilistic or statistics models (see [10] for a survey on this). Also due to the growth of online communities the use of different metrics has become an issue (see for example the reputation scores that eBay uses [1]). Flow models such as Advogato's reputation system [12] or Appleseed [16,17] use trust transitiveness. In these type of systems the reputation of a participant increases as a function of incoming flow and decreases as a function of ongoing flow.

There are many different trust models in the literature. The model we present in this paper is a graph-based model that allows us to represent trust paths as matrices. Our intention is to characterize trust metrics that are more suitable to be used in any given case, depending on the nature of the system, its properties, etc. As a novelty we propose the definition of a *trust function* that allows us to do this. A classification of trust metrics has been done in [17] but more oriented to the semantic web environment.

The paper is organized as follows. In Section 2 we outline how trust can be modelled as a graph and give some definitions. These definitions will be meaningful for Section 3 where we introduce our trust evaluation. Those definitions are going to be used for the instantiations of different operators in Section 4. Section 5 concludes the paper and outlines the future work.

2 A Graph Based Model of Trust

Trust in a virtual community can be modelled using a graph where the vertices are identified with the entities of the community and the edges correspond to trust relationships between entities. As we mentioned before, trust can be defined as the level of confidence that an entity s places on another entity t for performing a given task in a proper and honest way. The confidence level may vary depending on the task. Assuming that the level of confidence is a real number and that for each task there is only one trust value associated in our reasoning system, the trust graph is a weighted digraph.

Let us consider different tasks in our system. The trust graph will be a labelled multi digraph, i.e. there can be more than one edge from one particular vertex to another, where the label of each edge is compounded of a task identifier and the confidence level associated to it. That graph can also be modelled using a labelled digraph in which the labels consist of a sequence of labels of the previous type, each one corresponding to one edge of the multigraph. In this scenario we can distinguish two cases: (1) The simplest case where only one task is considered and (2) the average case where more than one task is considered.

The average case is quite easy to manage. For a fixed task identifier, we obtain a simple trust graph that can be inspected using techniques for the simplest case. The problem arises when there are dependencies among tasks. This could imply that implicit trust relationships can be found in the graph. An implicit trust relationship is derived from another one by applying some task dependency. For example, we can consider two dependent tasks, "Reading a file" and "Overwriting a file". Obviously they are trust dependant tasks, as trusting someone to overwrite some file should imply trusting him for reading that file too.

Those implicit trust relations depend on the kind of trust dependability that we allow in our system. The dependability rules have to be taken into account when reducing the trust graph for a given task. The dependency among tasks that we use in this paper is inspired in the definitions of the syntax of the *RT* framework, a family of Role-based Trust management languages for representing policies and credentials in distributed authorization [14]. In this work the authors define four different types of relationships among roles. If the relationships in the model are simple, these relationships can be modelled by using a partial order. This is the case for our purpose in this paper, a model of tasks, which are quite an objective concept. Next we will give some definitions.

Definition 1 (Trust Domain). *A trust domain is a partially ordered set* $(TD, <, 0)$ *where every finite subset of* TD *has a minimal element in the subset and* 0 *represents the minimal element of* TD.

Each entity in the system makes trust statements about the rest of the entities, regarding the task considered for each case. Those trust statements are defined as follows,

Definition 2 (Trust Statement). *A trust statement is an element* $(Trustor, Trustee, Task, Value)$ *in* $E \times E \times T \times TD$ *where,* E *is the set of all entities in the system;* T *is a partially ordered set representing the possible tasks, where the order established on tasks is* $\preceq$*; and* TD *is a Trust Domain.*

Let $G \subset E \times E \times T \times TD$ be a set of trust statements, and let x_0 be a fixed task in T, then G_{x_0} is defined as the set of trust statements of G such that the corresponding task is placed in an upper position in the task hierarchy, i.e.,

$$G_{x_0} = \{(s,t,x_0,v) \in E \times E \times T \times TD \text{ such that there exists } x \in T \text{such that } (s,t,x,v) \\ \in G \text{ and } x_0 \preceq x\}$$

Let now s_0 and t_0 be two fixed entities, then we can filter G in order to obtain a new set, G^{s_0,t_0}, as the trust statements of G such that they are part of a path from s_0 to t_0. We can combine the two filtering methods together to obtain a new set, $G^{s_0,t_0}_{x_0} = G^{s_0,t_0} \cap G_{x_0}$. We will see what these two sets are useful for in Section 3.

3 Trust Evaluations

If we want to establish trust between two entities in a system this trust should be measured somehow. A simple way to measure trust could be established by using a binary discrete model where the trust values are set as *a lot of trust*, for a very trusted entity, or *very little trust* if the trust placed in the entity measured is very low. More complicated systems could use integer numbers (Advogato's trust metric or FreeHaven [2]) or real numbers ([3,15]).

A *trust evaluation* or trust metric is a function such that given a trust graph, G, and two entities s and t, called the source and the target of trust respectively (trustor and trustee are alternative names for those entities) returns the level of trust or confidence that s places on t.

As the same entities can be trusted in different ways depending on the task to perform, this function also takes into account as a parameter the task we are referring to, in case there is more than one.

3.1 Trust Functions

Definition 3 (Trust Evaluation). *A trust evaluation for a trust graph G is a function* $\mathscr{F}_G : E \times E \times T \longrightarrow TD$*, where E, T and TD are the sets mentioned in Definition 2.*

We say that a trust evaluation is *local* if for any tuple $(s,t,x) \in E \times E \times T$, $\mathscr{F}_G(s,t,x) = \mathscr{F}_{G_x^{s,t}}(s,t,x)$, i.e., only those trust statements in $G_x^{s,t}$ are relevant for the evaluation.

In this work we focus on local trust evaluations, in particular on those trust evaluations that can be decomposed in two elemental functions: the Sequential Trust Function and the Parallel Trust Function. By decomposed functions we mean that the trust evaluation is computed by applying the Parallel Trust function to the results of applying the Sequential Trust Function over all the paths connecting two given entities.

Definition 4 (Sequential Trust Function). *A sequential trust function is a function,* $f : \bigcup_{n=2}^{\infty} \overbrace{TD \times \cdots \times TD}^{n} \longrightarrow TD$*, that calculates the trust level associated to a path or chain of trust statements, such that* $f(v_1,\ldots,v_n) = 0$ *if, and only if,* $v_i = 0$ *for any* $i \in \{1,\ldots,n\}$*, where* $v_i \in TD$ *and TD is a trust domain.*

Each path of trust statements in G is represented as the chain, $t_1 \xrightarrow{v_1} t_2 \xrightarrow{v_2} \cdots \xrightarrow{v_{n-1}} t_n \xrightarrow{v_n} x_{n+1}$, where t_i are entities in E and v_i are respectively the trust values associated to each statement.

The sequential trust function, f, may verify some of the following properties:

- Monotony (Parallel Monotony): $f(v_1,\ldots,v_n) \leq f(v'_1,\ldots,v'_n)$ if $v_i \leq v'_i$ for all $i \in \{1,\ldots,n\}$.
- Minimality: $f(v_1,\ldots,v_n) \leq min(v_1,\ldots,v_n)$
- Sequential monotony: $f(v_1,\ldots,v_{n-1},v_n) \leq f(v_1,\ldots,v_{n-1})$
- Preference Preserving: $f(v_1,\ldots,v_i,\ldots,v_j,\ldots,v_n) < f(v_1,\ldots,v_j,\ldots,v_i,\ldots,v_n)$ if $v_i < v_j$.
- Recursion: $f(v_1,\ldots,v_n) = f(f(v_1,\ldots,v_{n-1}),v_n)$

When defining a recursive sequential function we have to take into account that it is enough to define it over pairs of elements in TD, since by applying the recursion property we could obtain the value of the function for any tuple.

We call *generator sequential function or sequential operator* to the function f restricted over the domain $TD \times TD$. We represent it by $\odot$. Thus,

Definition 5 (Sequential Operator). *A Sequential Operator or Generator Sequential Function is defined as a function* $\odot : TD \times TD \longrightarrow TD$ *such that* $a \odot b = 0$ *if and only if* $a = 0$ *or* $b = 0$*.* $\odot(a,b)$ *or* $a \odot b$ *are used indistinctively for representing the same, whatever is more convenient.*

Given a recursive sequential function, f, the associated sequential operator $\odot_f$, can be defined as $a \odot b = f(a,b)$. Viceversa, given a sequential operator, the recursive inference sequential function can be defined as $f_\odot(z_1,\ldots,z_{n-1},z_n) = f_\odot(z_1,\ldots,z_{n-1}) \odot z_n$.

Note that a recursive sequential function verifies the reference preserving property only if the associated sequential operator, $\odot_f$, is not commutative.

Moreover, if $a \odot b \leq min(a,b)$, for any a and b, we could conclude that f verifies the minimality property.

Definition 6 (Parallel Trust Function). *A parallel trust function is used to calculate the trust level associated to a set of paths or chains of trust statements. It is defined as,* $g : \bigcup_{n=2}^{\infty} \overbrace{TD \times \cdots \times TD}^{n} \longrightarrow TD$*, where* TD *is a trust domain and*

1. $g(z_1,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_n) = g(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n)$ *if* $z_i = 0$
2. $g(z) = z$

g may verify the following desirable properties:

- Idempotency, $g(z,z,\ldots,z) = z$.
- Monotony, $g(z_1,z_2,\ldots,z_n) \leq g(z'_1,z'_2,\ldots,z'_n)$ If $z_i \leq z'_i$ for all$i \in \mathbb{N}$.
- Associativity, $g(g(z_1,z_2),z_3) = g(z_1,z_2,z_3) = g(z_1,g(z_2,z_3))$.

The *generator parallel function*, or the *parallel operator*, $\oplus$ for the function g, is defined analogously as the operator $\odot$.

Definition 7 (Parallel Operator). *A Parallel Operator or Generator Parallel Function is defined as a function,* $\oplus : TD \times TD \longrightarrow TD$*, such that* $a \oplus 0 = 0 \oplus a = a$

We say that the two operators $\oplus$ and $\odot$ are distributive if $(a \oplus b) \odot c = (a \odot c) \oplus (b \odot c)$.

In the case where there are no cycles in the trust graph, the set of paths connecting two any given entities is finite. Then, given a sequential operator $\odot$ and a commutative parallel operator $\oplus$, i.e. $a \oplus b = b \oplus a$, the associated trust evaluation, $\widehat{\mathscr{F}_G}$, is defined as follows,

Definition 8. *Let* $S_x^{s,t}$ *be the set of all paths of trust statements for task* x *starting in* s *and ending in* t*. For each path* $p \in S_x^{s,t}$ *represented as* $s \xrightarrow{v_1} \cdots \xrightarrow{v_n} t$ *let* z_p *be* $v_1 \odot \cdots \odot v_n$*, then* $\widehat{\mathscr{F}_G}(s,t,x)$ *is defined as* $\bigoplus_{p \in S_x^{s,t}} z_p$.

Given a fixed sequential operator, for any parallel operator that verifies idempotency and monotony properties then, $z_* = min_{p \in S_x^{s,t}} z_p \leq \bigoplus_{p \in S_x^{s,t}} z_p \leq max_{p \in S_x^{s,t}} z_p = z^*$. Therefore, the maximum and minimum possible trust values associated to a path from s to t are the upper and lower bounds for the trust evaluation $\widehat{\mathscr{F}_G}$.

Fortunately, we do not need to compute the trust values of each path in order to compute those bounds, i.e. z_* and z^* . In this case we can use an algorithm, adapted from the Dijkstra algorithm [8], to find for example, the maximum trust path from a fixed entity s to any other entity on the system. The minimum trust path can be computed in an analogous way.

This is a particular case of a trust evaluation where we use the maximum function as a parallel function. Unfortunately we can not generalize this kind of algorithms for other combinations of parallel and sequential functions as it heavily relies on the properties of the max and min functions.

3.2 Performing Trust Computations Using Matrices

Let us first assume the case where there are no cycles in the trust graph. We could model the trust network as a matrix, A where each element a_{ij} represents the trust level that node i places on node j. If we replace the scalar addition and multiplication in matrices by the operators $\oplus$ and $\odot$ respectively, then by iterating powers of the trust network we can compute the node to node trust values of the network. Thus, the trust evaluation could be defined through $\odot$ and $\oplus$ applying the generalized matrix product algorithm. It is then defined as $A \otimes B = \bigoplus_{k=1}^{n}(a_{ik} \odot b_{kj})$.

The generalized product is associative from the left hand side, i.e., $A \otimes B \otimes C = (A \otimes B) \otimes C$. Therefore, we can define the generalized power of A as $A^{(n)} = \bigotimes_{k=1}^{n} A$.

Last, we can define the operator $\oplus$ over matrices as $A \oplus B = (a_{ij} \oplus b_{ij})$, which can be used as the summation of matrices. Then, given a matrix A of order n, the matrix $\widehat{A}$ can be defined as $\widehat{A} = \bigoplus_{k=1}^{n} A^{(n)}$.

These definitions become more relevant when the aforementioned functions are distributive and associative in the case of the parallel function, and recursive in the case of the sequential function. However, they are still valid if these properties do not hold, although they may not be that meaningful.

Next, we will show that under the conditions mentioned above, the element (i,k) in the matrix $\widehat{A}$ is the value of the trust evaluation of all the trust paths from i to j. In particular, the first row in this matrix will give us the distribution of trust in the system.

First, we will show that the kth generalized power of the trust matrix A, $A^{(k)}$ contains the trust through paths of length k. We will prove this by induction where the base case holds by the definition of matrix A.

We assume that for $k \in \mathbb{N}$ (i,j) in $A^{(k)}$, $a_{ij}^{(k)}$, represents the value of the trust evaluation of all the trust paths of length k. We will then show that this is also the case for length $k+1$.

Let $C_{ij} = \{c_{ij}^{1}, \dots, c_{ij}^{m_{ij}}\}$ be the set of all the paths of trust values from all the paths of length k from i to j. Then since $A^{(k+1)} = A^{(k)} \otimes A$, each element of the matrix can be obtained by using the function g as $a_{ij}^{k+1} = \bigoplus_{l=1}^{n}(a_{il}^{(k)} \odot a_{lj}) = g(c_{i1}^{1} \odot a_{1j}, \dots, c_{i1}^{m_{i1}} \odot a_{1j}, \dots, c_{in}^{1} \odot a_{nj}, \dots, c_{in}^{m_{in}} \odot a_{nj})$.

Let now $c \equiv i \xrightarrow{z_1} c_1 \xrightarrow{z_2} \cdots c_n \xrightarrow{z_k} l \xrightarrow{z_{k+1}} j$ be a $k+1$ length path from i to j. This path can also be expressed as the path c' of length k from i to l linked with the path from l to j, (l,j). Therefore, since the sequential function is recursive, any path of length $k+1$ from i to j can be obtained adding a link to any path of length k. Thus, a_{ij}^{k+1} represents the value of the trust evaluation of all the paths of length $k+1$ from i to j. The number of elemental operations (sequential and parallel operations) for computing $\widehat{A}$ is

$$\frac{n(n^3 - 2n^2 - n + 2)}{12} \qquad (1)$$

The important issue is that the order of operations is $\mathscr{O}(n^4)$.

3.3 The Problem with Cycles

If there are cycles in the trust graph the previous definitions and algorithm are not valid, therefore we need and extra algorithm to compute the trust values for this case. In fact, the new algorithm we are going to introduce is only needed when we are computing the trust value of a node involved in a cycle.

The key aspect of this new algorithm is to remove redundant graphs from the trust graph in such a way that the set of paths connecting two given entities remains finite.

Let i and j be two nodes in the system and m a natural number, then we can define the set S_{ij}^m as the subset of the permutation group S_n containing all the cycles, σ, of length $m+1$ such that $\sigma^m(i) = j$.

The cardinality of the set S_{ij}^m is $\mid S_{ij}^m \mid = \frac{(n-2)!}{(n-m-1)!}$. Thus, the number of elements is of the order $\mathscr{O}(n^{m-1})$.

The intuition behind the new algorithm is the same as for the previous one except by the fact that we only modify the way we compute the elements of the matrices $A^{(m)}$. In the case of nodes which are not involved in any cycle the two algorithms provide the same result.

For the new algorithm $a_{ij}^{(1)} = a_{ij}$ and $a_{ij}^{(m)} := \sum_{\sigma \in S_{ij}^m} a_{i,\sigma(i)} \odot \cdots \odot a_{\sigma^{m-1}(i)j}$.

The number of parallel operations performed by this algorithm is $(n^2)\frac{(n-2)!}{(n-m-1)!}$. This number is of the order of $\mathscr{O}(n^{m+1})$. As the length of the trust path is m, each of the components in the previous operations need $m-1$ sequential operations therefore the total number of sequential operations is of the order of $\mathscr{O}((m-1)n^{m+1})$. Thus, adding up these two number of operations, we can conclude that the total number of operations is $\mathscr{O}(mn^{m+1})$ only for the matrix $A^{(m)}$. Therefore, if we compute all the trust paths for all $m \in \mathbb{N}$ the amount of operations will grow enormously. If we compare this number with the number of operations in Equation 1 we can conclude that this latter number is much bigger for any $m > 3$.

As we can see by observing the number of operations for both algorithms, they are higher for the new algorithm, therefore it will be convenient to avoid cycles, if possible. We might need to apply some techniques for this, for example, we can include a timestamp in each trust statement and remove the problematic edges depending on when they were created.

4 Examples of the Model

The properties of the system are going to be derived from the properties of the operators $\odot$ and $\oplus$. Depending on these properties we could classify or outline the systems.

As we will only consider recursive functions we will deal directly with operators instead of functions. For simplicity and for showing the model purposes, we will consider the initial trust domain to be the interval $[0,1]$, where 0 stands for null trust and 1 for full trust. However, other more complex, non-scalar domains can be used.

We only consider two possibilities for the sequential operator: Product and Minimum; and for the parallel operators Maximum, Minimum and Mean. Regarding the sequential operators, their definitions are straightforward. Both of them trivially verify monotony, minimality and sequential monotony properties. The product also verifies what we could call “Strict Sequential Monotony”. This means that $v_1 \odot v_2 < 1$ if $v_2 < 1$. The preference preserving property does not hold for any of them.

Note that in order to be able to perform the computation of trust in a distributed manner we need the operators to be distributive. Then the problem of defining a parallel operator become harder as we have to make them compatible with the definition of the sequential operators.

4.1 Completing the Model Instances

In this section we will concentrate on parallel operators.

Maximum and Minimum. The Maximum and Minimum are two examples of parallel operators which verify the associativity property as well as idempotency and monotony properties. The Minimum operator however does not verify the definition of parallel operator strictly as the minimum of any set containing 0 will be 0. This problem can be solved by erasing the 0s of such a sets before applying the function. The resulting operator with this new domain is called $\oplus_{min^*}$.

$$v_1 \oplus_{min^*} v_2 := \begin{cases} min\{v_1, v_2\} & \text{if } v_1 \neq 0 \text{ and } v_2 \neq 0 \\ v_1 & \text{if } v_2 = 0 \\ v_2 & \text{if } v_1 = 0 \end{cases}$$

Both operators, $\oplus_{min^*}$ and $\oplus_{max}$ verify the distributivity property with respect to the sequential operator minimum, i.e.,

1. $(v_1 \oplus_{max} v_2) \odot_{min} \lambda = (v_1 \odot_{min} \lambda) \oplus_{max} (v_2 \odot_{min} \lambda)$
2. $(v_1 \oplus_{min^*} v_2) \odot_{min} \lambda = (v_1 \odot_{min} \lambda) \oplus_{min^*} (v_2 \odot_{min} \lambda)$

and distributivity with respect to the sequential operator product, i.e.,

1. $(v_1 \oplus_{max} v_2) \odot_{\cdot} \lambda = (v_1 \odot_{\cdot} \lambda) \oplus_{max} (v_2 \odot_{\cdot} \lambda)$
2. $(v_2 \oplus_{min^*} v_2) \odot_{\cdot} \lambda = (v_1 \odot_{\cdot} \lambda) \oplus_{min^*} (v_2 \odot_{\cdot} \lambda)$

As we mentioned in Section 3.1 the maximum and minimum functions are the upper and lower bounds of any parallel function that verifies the idempotency and monotony properties. The difference between the highest trust and the lowest trust will be the range of the variation of trust for any other election of the parallel operator. This range of values will give us an average of the deviation of trust.

Mean. The mean verifies idempotency and monotony properties but, however, does not verify the associativity property. We could solve this problem by using a different trust domain $\mathscr{T} := [0,1] \times \mathbb{N}$ and defining the operator as follows,

$$\begin{aligned} \oplus_{Mean*} : ([0,1] \times \mathbb{N}) \times ([0,1] \times \mathbb{N}) &\longrightarrow ([0,1] \times \mathbb{N}) \\ ((v_1, n), (v_2, m)) &\longmapsto \begin{cases} \left(\dfrac{v_1 \cdot n + v_2 \cdot m}{n+m}, n+m \right) & \text{if } v_1 \neq 0 \text{ and } v_2 \neq 0 \\ (v_1, n) & \text{if } v_2 = 0 \\ (v_2, m) & \text{if } v_1 = 0 \end{cases} \end{aligned}$$

where ‘$\cdot$’ is the usual product in $\mathbb{R}$.

With this definition the operator is associative and still verifies idempotency and monotony properties. The distributivity property only holds for the sequential operator product defined in the trust domain $[0,1] \times \mathbb{N}$ as $(v_1, n_{v_1}) \odot (v_2, n_{v_2}) = (v_1 \cdot v_2, n_{v_1} \cdot n_{v_2})$.

The generalized product of matrices can be applied to the combination $(\odot, \oplus_{Mean^*})$. This will allow us to calculate the mean of the trust values of any entity by using the matrix $\widehat{A}$ in the domain $[0,1] \times \mathbb{N}$. Note that the first component of a trust value in this domain represents the actual trust level, whereas the second one represents the number of paths considered for this computation.

4.2 Summary of the Examples

We have proposed the following combination of operators of our model based on the operators defined along the paper.

1. $(Min, Min*)$. In this case the minimum is the sequential function, *Min*, and the minimum of the non-null elements, *Min*∗, is the parallel function.
2. (Min, Max). The function minimum is the sequential function and the maximum, *Max* is the parallel function.
3. $(Product, Min*)$. The function product is the sequential function and *Min*∗ is the parallel function.
4. $(Product, Max)$. The function product is the sequential function and *Max* is the parallel function.
5. $(Product, Mean*)$. The product is the sequential function and the mean of the non-null elements is the parallel function. The trust domain in this case is not the interval $[0,1]$ as in the other cases but $[0,1] \times \mathbb{N}$.

5 Conclusions and Future Work

We have introduced a general model of trust that splits the process of computing trust in two sub-processes. In order to do this we have introduced the sequential and the parallel operators, which will be the parameters used in our model, together with a trust domain that can be any set used for defining trust values. Depending on those parameters, the trust values computed by the model will vary.

We assume that trust between entities is directional and can be partial. By partial we mean that it can be related to some specific task but not all of them, i.e. one can trust someone else to perform a certain task in an honest manner but not for the other tasks. Those tasks can be related, therefore it could be useful to order or classify them in a hierarchical diagram, in a way that trusting some entity for a certain task will imply trusting this entity for all the tasks that are lower in the hierarchy. How to order tasks is out of the scope of this paper although we consider it is an important aspect of the model that remains for future work.

We have defined the model by using a labelled trust graph where the label is of the form (t, x_0), where t is the value of trust and x_0 is a task. When performing trust evaluations, we set a fixed task in order to be able to remove this parameter from the labels. The resulting weighted graph is then processed by using the sequential and parallel operators. We have also presented some sample uses of our model with simple operators and a simple trust domain (the interval [0,1]) to show that the model is useful when defining new trust evaluations.

In the future we intend to investigate the possibility of using different trust domains other than $[0,1]$. We are particularly interested in investigating the trust domain that the Subjective Logic by Jøsang uses [9]. Analysis of a trust network using subjective logic have been already carried out [11]. Our intention is to analyze the operators defined for it them according to the properties that we have defined and therefore, how our model could be suitable for them. We also intend to define new operators and study the properties that they may verify.

References

1. http://www.ebay.com
2. http://www.freehaven.net/
3. http://www.openprivacy.org/
4. Blaze, M., Feigenbaum, J., Keromytis, A.D.: KeyNote: Trust Management for Public-Key Infrastructures (position paper). In: Christianson, B., Crispo, B., Harbison, W.S., Roe, M. (eds.) Security Protocols 1998. LNCS, vol. 1550, pp. 59–63. Springer, Heidelberg (1999)
5. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: IEEE Symposium on Security and Privacy (1996)
6. Burrows, M., Abadi, M., Needham, R.M.: A Logic of Authentication. ACM Trans. Comput. Syst. 8(1), 18–36 (1990)
7. Chu, Y.H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: REFEREE: Trust Management for Web Applications. Computer Networks and ISDN Systems 29, 953–964 (1997)
8. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
9. Jøsang, A., Ismail, R.: The Beta Reputation System. In: 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, Bled, Slovenia (June 2002)
10. Jøsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems 43, 618–644 (2007)
11. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: ACSC 2006: Proceedings of the 29th Australasian Computer Science Conference, 2006, Darlinghurst, Australia, pp. 85–94. Australian Computer Society, Inc. (2006)
12. Leiven, R.: Attack Resistant Trust Metrics. PhD thesis, University of California, Berkeley (2003)
13. Leiven, R., Aiken, A.: Attack-Resistant Trust Metrics for Public Key Certification. In: Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, USA (January 1998)
14. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust management framework. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, pp. 114–130. IEEE Computer Society Press, Los Alamitos (May 2002)
15. Marsh, S.: Formalising Trust as a Computational Concept. PhD thesis, Department of Computer Science and Mathematics, University of Stirling (1994)
16. Ziegler, C.N., Lausen, G.: Spreading Activation Models for Trust Propagation. In: IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE 2004), Taipei (March 2004)
17. Ziegler, C.-N., Lausen, G.: Propagation Models for Trust and Distrust in Social Networks. Information Systems Frontiers 7(4-5), 337–358 (2005)

Patterns and Pattern Diagrams for Access Control

Eduardo B. Fernandez[1], Günther Pernul[2], and Maria M. Larrondo-Petrie[2]

[1] Florida Atlantic University, Boca Raton, FL 33431, USA
ed@cse.fau.edu, petrie@fau.edu
[2] University of Regensburg, Universitätsstraße 31, Regensburg, Germany
guenther.pernul@wiwi.uni-regensburg.de

Abstract. Access control is a fundamental aspect of security. There are many variations of the basic access control models and it is confusing for a software developer to select an appropriate model for her application. The result in practice is that only basic models are used and the power of more advanced models is thus lost. We try to clarify this panorama here through the use of patterns. In particular, we use pattern diagrams to navigate the pattern space. A pattern diagram shows relationships between patterns and we can see how different models relate to each other. A subproduct of our work is the analysis of which patterns are available for use and which need to be written. Pattern maps are also useful to perform semi-automatic model transformations as required for Model-Driven Development (MDD). The idea is to provide the designer of a secure system with a navigation tool that she can use to select an appropriate pattern from a catalog of security patterns. We also indicate how to compose new access control models by adding features to an existing pattern and how to define patterns by analogy.

1 Introduction

The development of secure systems requires that security be considered at all stages of design, so as to not only satisfy their functional specifications but also satisfy security requirements. Several methodologies that apply security at all stages have been proposed [1], [2], [3]. Some of these methodologies start from use cases and from them a conceptual model is developed. Security constraints are then defined in the conceptual model. To do this we need high-level models that represent the security policies that constrain applications. One of the most fundamental aspects of security is access control.

Although there are only a few basic access control models, many varieties of them have been proposed. It is confusing for a software developer to select an appropriate model for her application. Access control models generally represent a few types of security policies, e.g. “rights are assigned to roles”, and provide a formalization of these policies using some ad hoc notation. Four basic access control models are commonly used and they may be extended to include content and context-based access control, delegation of rights, hierarchical structuring of subjects (including roles), objects, or access types [4], temporal constraints, etc. Access control models can be defined for different architectural levels, including application, database systems,

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 38–47, 2008.

operating systems, and firewalls [5]. Some of them apply to any type of systems while some are specialized, e.g. for distributed systems.

Access control models fall into two basic categories: Mandatory Access Control (MAC), where users' rights are defined by administrators and data may be labeled to indicate its sensitivity, and Discretionary Access Control (DAC), where users may administer the data items they create and own. In a MAC model, users and data are classified by administrators and the system applies a set of built-in rules that users cannot circumvent. In a DAC model, there is no clear separation of use and administration; users can be owners of the data they create and act as their administrators. Orthogonal to this classification, there are several models for access control to information that differ on how they define and enforce their policies [6], [7]. The most common are:

- The *Multilevel model* organizes the data using security levels. This model is usually implemented as a mandatory model where its entities are labeled indicating their levels. This model is able to reach a high degree of security, although it can be too rigid for some applications. Usually, it is not possible to structure the variety of entities involved in complex applications into strictly hierarchical structures.
- The *Access Matrix* describes access by subjects (actors, entities) to protected objects (data/resources) in specific ways (access types) [8], [6], [7]. It is more flexible than the multilevel model and it can be made even more flexible and precise using predicates and other extensions. However, it is intrinsically a discretionary model in which users own the data objects and may grant access to other subjects. It is not clear who owns the medical or financial information and the discretionary property reduces security. This model is usually implemented using Access Control Lists (lists of the subjects that can access a given object) or Capabilities (tickets that allow a process to access some objects).
- *Role-Based Access Control* (RBAC), collects users into roles based on their tasks or functions and assigns rights to each role [9]. Some of these models, e.g. [10], [11], have their roles structured as hierarchies, which may simplify administration. RBAC has been extended and combined in many ways.
- *Attribute-Based Access Control* (ABAC). This model controls access based on properties (attributes) of subjects or objects. It is used in environments where subjects may not be pre-registered [12].

While these basic models may be useful for specific domains or applications, they are not flexible enough for the full range of policies present in some of these applications [5], [4]. This is manifested in the large variety of ad hoc RBAC variations that have been proposed; most of which add specialized policies to a basic RBAC model. For example, some models have added content or context-dependent access [13], delegation [14], task-based access [15], and relationships between role entities [16]. All these models effectively incorporate a set of built-in access control policies and cannot handle situations not considered by these policies, which means that a complex system may need several of these models for specific users or divisions.

All these models present a bewildering set of options to the designer, who has problems deciding which model to use. The result in practice is that only basic models are used and the power of more advanced models is thus lost. We try to clarify this panorama here through the use of patterns. In particular, we use pattern diagrams to navigate the pattern space. A pattern diagram shows relationships between patterns (represented by rectangles with rounded corners). A subproduct of our work is the analysis of which patterns are available for use and which need to be written. Pattern maps are also useful to perform semi-automatic model transformations as required for Model-Driven Development (MDD). They can serve as metamodels of possible solutions being added at each transformation.

A pattern is an encapsulated solution to a recurrent problem in a given context. In particular, a security pattern describes a mechanism that is a solution to the problem of controlling a set of specific threats [17]. This solution is affected by some forces and can be expressed using UML class, sequence, state, and activity diagrams. A set of consequences indicate how well the forces were satisfied; in particular, how well the attacks can be handled by the pattern. A study of the forces and consequences of a pattern is important before their final adoption; however, a good initial pattern selection is fundamental to avoid a lengthy search through textual pattern descriptions. A requirement for a pattern is that the solution it describes has been used in at least three real systems [18], [19]. This is consistent with the idea of patterns as best practices. However, a pattern can also describe solutions that have not been used (or have been used only once) but appear general and useful for several situations. Because of this, we include here both types: good practices patterns and useful solutions patterns. In fact, as mentioned above, many models have never been used in practice.

We do not attempt to be exhaustive because the quantity of models is too large, some are just simple variations of others, and some appear to have scarce practical value. How exhaustive the catalog needs to be depends on the variety of applications to be handled. The idea is to provide the designer of a secure system with a way to organize a navigation tool that she can use to select an appropriate pattern from a catalog of security patterns. We also indicate how to compose new access control models by adding features to an existing pattern and how to define patterns by analogy.

Section 2 presents the use of pattern diagrams to relate access control models. Section 3 discusses how patterns can be defined at different levels of abstraction. Section 4 considers how to grow new models from existing ones, while Section 5 shows how to obtain models by analogy. We end with some conclusions.

2 Pattern Diagrams for Access Control Patterns

Access control models have two aspects: a definition of a set of rules specifying the valid accesses (some of them may be implied by other rules), and an enforcement mechanism that intercepts access requests from users or processes and determines if the request is valid. The main difference between models is on the way they define their rules, so it makes sense to separate the patterns for enforcement mechanisms; that is, we should provide separate patterns for rules and for enforcement mechanisms. Typically, there is much less variety in the enforcement mechanism: it intercepts requests and makes a decision based on the rules. As an illustration of how pattern diagrams can put models in perspective, Figure 1 shows some variations of access control models. One of the first access control models was the access matrix. The basic access matrix [7] included the tuple {s,o,t}, where s indicates a subject or active entity, o is the protected object or resource, and t indicates the type of access permitted. [8] proved security properties of this model using the so-called HRU (Harrison-Ruzzo-Ullman) model. In that model users are allowed to delegate their rights (discretionary property, delegatable authorization), implying a tuple {s,o,t,f}, where f is a Boolean copy flag indicating if the right is allowed to be delegated or not. A predicate was added later to the basic rule to allow content-based authorization [20], becoming {s,o,t,p,f}, where p is the predicate (the predicate could also include environment variables). Patterns for the basic rule and for the one with tuple {s,o,t,p,f} were given in [21], [17]. The rule could also include the concept of Authorizer (a), becoming {a,s,o,t,p,f} [22] (Explicitly Granted Authorization). RBAC [9] can be considered a special interpretation of the basic authorization model, where subjects are roles instead of individual users. We presented two varieties of RBAC patterns in [21] and [17]. We combined it with sessions in [23] (The double-lined patterns of Figure 1). Several variations and extensions of these models have appeared. We presented a variation called Metadata-Based Access Control, which later we renamed Attribute-Based Access Control (ABAC) [12].

Figure 1 assumed that we started from some known models. Figure 2 starts from the basic components of access control to provide a more general approach to developing access control models (we did not show the labels of the links for simplicity). This diagram can be the starting point that allows a designer to select the type of access control he needs in his application. Once this abstract level is clear, we need to go to a software-oriented level where we can choose more specific approaches. The center of this diagram is Policy-Based Access Control (PABC) which indicates that the rules represent access policies, which are in turn defined by a Policy pattern. The Policy-Based Access Control pattern decides if a subject is authorized to access an object according to policies defined in a central policy repository. The enforcement of these policies is defined by a Reference Monitor pattern. Depending on its administration, PABC can be MAC or DAC. XACML is a type of PBAC oriented SOA [24], shown here as two patterns for its aspects of rule definition and evaluation. Policies can be implemented as Access Control Lists (ACLs) or Capabilities. The NIST pattern is a variety of RBAC discussed in Section 4. The reference Monitor may use a Policy Enforcement Point (PEP), a Policy Definition Point (PDP), and other patterns to describe the administrative structure of enforcement [24]. The Access Matrix can be extended with predicates or a copy flag and both can be used in another variation of this model.

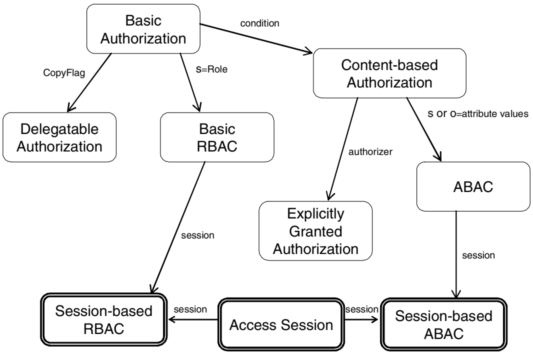


Fig. 1. Relationships between access control patterns

3 Levels of Abstraction

Models and their patterns may correspond to different abstraction levels; for example, the concept of session is a concept at a lower level than RBAC because it indicates an abstract mechanism to restrict the concurrent use of some rights (which may produce conflicts of interest). It is possible to make explicit in the pattern diagram the abstraction level of the patterns Figure 3 shows these levels explicitly, showing the Policy Model, Policy Rule,and reference Monitor at the highest level, while PBAC and the Session-Based Reference Monitor are more implementation-oriented. Many times we do not emphasize this separation; however, when we deal with different architectural levels this separation is important; for example, the implementation of access control at the file system level is quite different from authorization rules at the application level. Figure 3 also shows how some of the patterns of Figure 1 could have been found starting from the components of Figure 2: We can define a Session-Based Reference Monitor that requires the concept of Access Session to delimit the rights of the user. This figure also emphasizes the separation of Policy Model and Policy Rules, not shown in Figure 2.

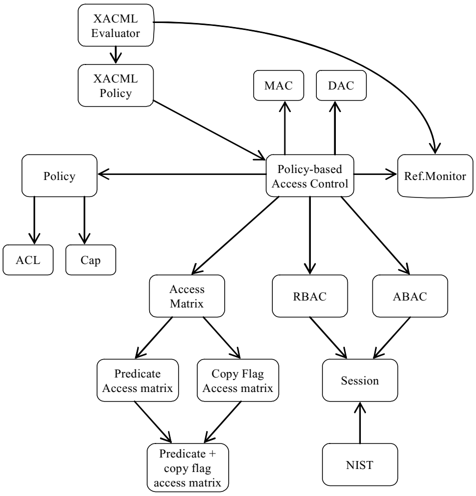


Fig. 2. A classification of access control patterns

4 Using the Diagrams

Let's consider a financial application. The threats to this system have been identified and indicate that we need access control for classes Account, Transaction, and Customer. The designer refers to a diagram such as Figure 2 and starts by choosing PBAC because we need to apply banking policies to control access; for example, the owner of the account has the right to deposit and withdraw money from the account. The designer then chooses the Access Matrix model because access to accounts is given to individuals, not roles. As owners should only access their own

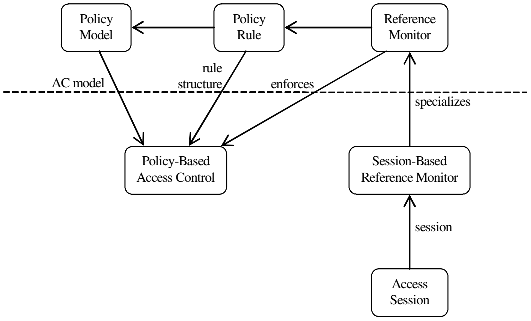


Fig. 3. Deriving specialized patterns from abstract models

accounts, we need predicates in the Access Matrix. Later, when the software architecture is defined, the designer decides to use web services, because ATMs and branch offices make this application highly distributed. Since any PBAC can be implemented using XACML, the designer implements a Predicate Access Matrix using XACML.

5 Growing New Models

To apply this design approach we need good pattern catalogs. In this and the next section we see two approaches to develop catalogs. Each pattern can be augmented with new features to produce a new model with larger functionality. Figure 4 shows the basic RBAC pattern, where users are assigned to roles and roles are given rights. The National Institute of Standards and Technology (NIST) RBAC pattern follows the NIST standard [21] and allows access to resources based on the role of the subject and adds several functions to the basic model. The NIST standard also adds a linear role hierarchy. A subject may have several roles. Each role collects the rights that a user can activate at a given moment (execution context), while a session controls the way of using roles and can enforce role exclusion at execution time. Strictly, the NIST standard allows only linear role hierarchies, while Figure 5 shows recursive hierarchies (described by the Composite pattern [19]). Figure 5 shows also object/resource recursive hierarchies. Similarly, we can separate administrative roles and rights and give roles to groups of users for example. The idea is that new functions require only adding more classes and their corresponding associations. In fact, the pattern chains in the models of Figures 1 and 2 can be obtained in this way; for example, in Figure 2 we added a session to RBAC to obtain an NIST RBAC pattern.

Combining this idea with architectural levels we can define different variations of these patterns intended for more specialized uses. For example, [25] shows an RBAC model where access to objects is performed through views. Specific views carry sets of rights as in the case of database systems. We can also formalize these models by adding OCL constraints in each pattern. The constraints may make functionality more

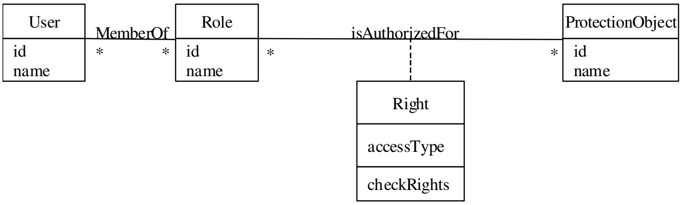


Fig. 4. Basic RBAC pattern

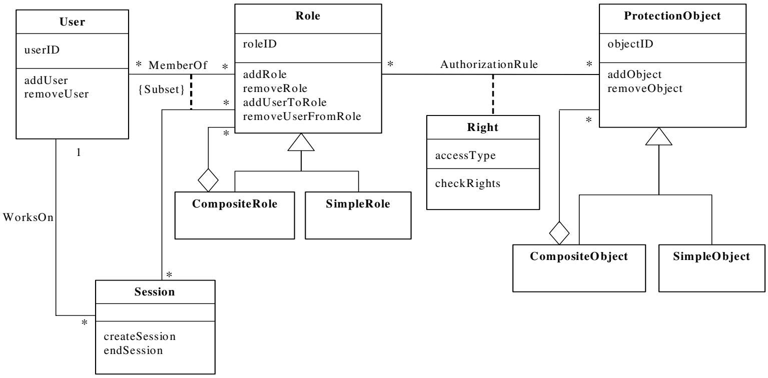


Fig. 5. Class model for RBAC with sessions and role/resource hierarchies

specific, for example, by defining constraints among variables, or may define pre- or post-conditions for the pattern. As each model is derived from a simpler model, it is easy for the designer to see the differences and possibilities of each model.

6 Finding Access Control Models by Analogy

Analogy is another way to derive models from existing models [26].A model for medical records such as the one in Figure 6 represents typical policies used in HIPAA and other regulations. In particular, this model represents the policies:

- A **Patient** role that has the rights to read his own record and authorize the use of this record.
- A **Doctor** role showing that a given doctor may act as custodian for a patient record.
- The **Medical Record**, that includes the constraint that any reading of a record by persons other than health providers, must be notified to the corresponding patient.
- Specific medical records may be associated (linked) with other records to describe, for example, family relationships and physical contact, needed to trace genetic or infectious diseases.

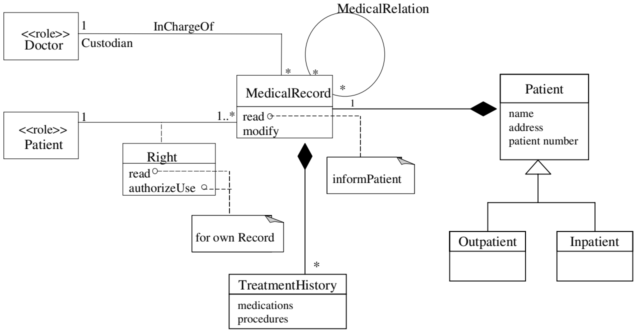


Fig. 6. A model for medical policies

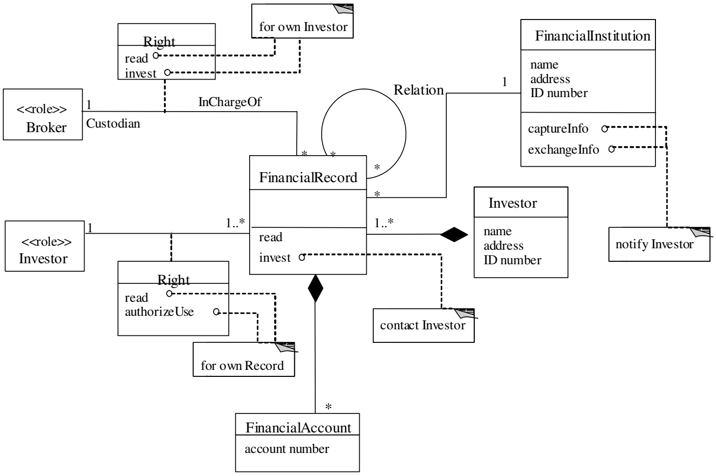


Fig. 7. A pattern that includes some of the Sarbanes Oxley policies

If we need a model for the Sarbanes/Oxley regulations, we can make the following analogies: Patient—Investor; Doctor—Broker; Hospital—Financial Institution; Medical Record—Financial Record. This leads to Figure 7, which is basically the same structure although the behavior semantics of some classes may be different.

7 Conclusions

We have tried to unify the many access control model varieties by using patterns. We believe that this perspective can help developers to align their needs with the selection of appropriate access control models. The selected access control pattern not only guides the conceptual security of the application but later it also guides the actual implementation of the model.

We can navigate the pattern diagram because (as shown in Section 4) patterns are composable with features, i.e. adding a feature (perhaps embodied by another pattern) produces a new pattern with extra features. If we have a pattern diagram we can navigate in it to find an appropriate pattern

Using this approach we can not only clarify the relationships of access control models but it has led us also to discover the need for new security patterns: Subject, Object, Labeled Security, DAC, MAC. Access control patterns give us also the possibility of evaluating commercial products: we can see if the product contains the corresponding pattern in its design.

We are working on the necessary environment to use this approach, including:

- A complete catalog of security patterns including many varieties of access control models to let designers find the required solution to each security situation
- A classification of patterns according to their security concerns and architectural level. We proposed such a classification in [27].
- A tool incorporating pattern classifications and a catalog. Pattern maps would make use of pattern classifications to select which patterns to present to a designer; for example, operating system-level patterns to a system designer.
- A standard representation of security patterns. This is important for implementing tools and for a more widespread use of security patterns.

References

1. Fernandez, E.B., Larrondo-Petrie, M.M., Sorgente, T., VanHilst, M.: A methodology to develop secure systems using patterns. In: Mouratidis, H., Giorgini, P. (eds.) Integrating security and software engineering: Advances and future vision, pp. 107–126. IDEA Press (2006)
2. Mouratidis, H., Jurjens, J., Fox, J.: Towards a Comprehensive Framework for Secure Systems Development. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 48–62. Springer, Heidelberg (2006)
3. Yoshioka, N.: A development method based on security patterns. Presentation, NII, Tokyo (2006)
4. De Capitani di Vimercati, S., Samarati, P., Jajodia, S.: Policies, models, and languages for access control. In: Bhalla, S. (ed.) DNIS 2005. LNCS, vol. 3433, pp. 225–237. Springer, Heidelberg (2005)
5. De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: Access control: principles and solutions. Software - Practice and Experience 33(5), 397–421 (2003)
6. Gollmann, D.: Computer Security. John Wiley & Sons, New York (2006)
7. Summers, R.C.: Secure Computing: Threats and Safeguards. McGraw-Hill, New York (1997)
8. Harrison, M., Ruzzo, W., Ullman, J.: Protection in Operating Systems. Communications of the ACM 19(8), 461–471 (1976)
9. Sandhu, R., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer 29(2), 38–47 (1996)
10. Sandhu, R., Bhamidipati, V., Munawer, G.: The ARBAC97 model for role-based administration of roles. ACM Transactions on Information and System Security 2(1), 105–135 (1999)

11. Thomsen, D., O'Brien, R.C., Bogle, J.: Role Based Access Control framework for network enterprises. In: Proc. of the 14th Annual Computer Security Applications Conference, pp. 50–58. IEEE Press, New York (1998)
12. Priebe, T., Fernandez, E.B., Mehlau, J.I., Pernul, G.: A pattern system for access control. In: Farkas, C., Samarati, P. (eds.) Research Directions in Data and Applications Security XVIII, Proc. of the 18th. Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Springer, New York (2004)
13. Chandramouli, R.: A Framework for Multiple Authorization Types in a Healthcare Application System. In: Proc. of the 17th Annual Computer Security Applications Conference, ACSAC, pp. 137–148. IEEE Press, New York (2001)
14. Zhang, L., Ahn, G.J., Chu, B.T.: A role-based delegation framework for healthcare systems. In: Proc. of the 8th ACM Symposium on Access Control Models and Technologies, SACMAT 2002, pp. 125–134 (2003)
15. Thomas, R.K.: Team-Based Access Control (TMAC): A primitive for applying role-based access controls in collaborative environments. In: Proc. of the 2nd ACM Workshop on Role-based access control, RBAC 1997, pp. 13–19 (1997)
16. Barkley, J., Beznosov, K., Uppal, J.: Supporting relationships in access control using Role Based Access Control. In: Proc. of ACM Role-Based Access Control Workshop, RBAC 1999, pp. 55–65 (1999)
17. Schumacher, M., Fernandez, E.B., Hybertson, D., Buschmann, F., Sommerlad, P.: Security Patterns: Integrating security and systems engineering. John Wiley & Sons, New York (2006)
18. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture. A System of Patterns, vol. 1. John Wiley & Sons, New York (1996)
19. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston (1994)
20. Fernandez, E.B., Summers, R.C., Coleman, C.B.: An authorization model for a shared data base. In: Proc. of the 1975 ACM SIGMOD International Conference, pp. 23–31 (1975)
21. Fernandez, E.B., Pan, R.: A pattern language for security models. In: Proc. of the 8th Pattern Languages of Programs Conference, PLOP 2001, pp. 11–15 (2001), http://jerry.cs.uiuc.edu/~plop/plop2001/accepted_submissions/PLoP2001/ebfernandezandrpan0/PLoP2001_ebfernandezandrpan0_1.pdf
22. Fernandez, E.B., Summers, R.C., Wood, C.: Database Security and Integrity (Systems Programming Series). Addison-Wesley, Reading (1981)
23. Fernandez, E.B., Pernul, G.: Patterns for Session-Based Access Control. In: Proc. of the Conference on Pattern Languages of Programs, PLoP 2006 (2006)
24. Delessy, N., Fernandez, E.B., Larrondo-Petrie, M.M.: A pattern language for identity management. In: Proceedings of the 2nd IEEE International Multiconference on Computing in the Global Information Technology, ICCGI 2007 (2007)
25. Koch, M., Parisi-Presicce, F.: Access Control Policy Specification in UML. In: Proc. of Critical Systems Development with UML, satellite workshop of UML 2002, TUM-I0208, pp. 63–78 (2002)
26. Fernandez, E.B., Yuan, X.: Semantic analysis pattern. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 183–195. Springer, Heidelberg (2000)
27. Fernandez, E.B., Washizaki, H., Yoshioka, N., Kubo, A., Fukazawa, Y.: Classifying security patterns. In: Zhang, Y., Yu, G., Bertino, E., Xu, G. (eds.) APWeb 2008. LNCS, vol. 4976, pp. 342–347. Springer, Heidelberg (2008)

A Spatio-temporal Access Control Model Supporting Delegation for Pervasive Computing Applications*

Indrakshi Ray and Manachai Toahchoodee

Department of Computer Science
Colorado State University
Fort Collins CO 80523-1873
{iray,toahchoo}@cs.colostate.edu

Abstract. The traditional access control models, such as Role-Based Access Control (RBAC) and Bell-LaPadula (BLP), are not suitable for pervasive computing applications which typically lack well-defined security perimeters and where all the entities and interactions are not known in advance. We propose an access control model that handles such dynamic applications and uses environmental contexts to determine whether a user can get access to some resource. Our model is based on RBAC because it simplifies role management and is the de facto access control model for commercial organizations. However, unlike RBAC, it uses information from the environmental contexts to determine access decisions. The model also supports delegation which is important for dynamic applications where a user is unavailable and permissions may have to be transferred temporarily to another user/role in order to complete a specific task. This model can be used for any application where spatial and temporal information of a user and an object must be taken into account before granting access or temporarily transferring access to another user.

1 Introduction

With the increase in the growth of wireless networks and sensor and mobile devices, we are moving towards an era of pervasive computing. The growth of this technology will spawn applications such as, the Aware Home [6] and CMU's Aura [7], that will make life easier for people. However, before such applications can be widely deployed, it is important to ensure that no authorized users can access the resources of the application and cause security and privacy breaches. Traditional access control models, such as, Bell-LaPadula (BLP) and Role-Based Access Control (RBAC), do not work well for pervasive computing applications because they do not have well-defined security perimeters and all the users and resources are not known in advance. Moreover, they do not take into account environmental factors, such as, location and time, while making access decisions. Consequently, new access control models are needed for pervasive computing applications.

In pervasive computing applications, the access decisions cannot be based solely on the attributes of users and resources. For instance, we may want access to a computer be

* This work was supported in part by AFOSR under contract number FA9550-07-1-0042.

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 48–58, 2008.

enabled when a user enters a room and it to be disabled when he leaves the room. Such types of access control can only be provided if we take environmental contexts, such as, time and location, into account before making access decisions. Thus, the access control model for pervasive computing applications must allow for the specification and checking of environmental conditions.

Pervasive computing applications are dynamic in nature and the set of users and resources are not known in advance. It is possible that a user/role for doing a specific task is temporarily unavailable and another user/role must be granted access during this time to complete it. This necessitates that the model be able to support delegation. Moreover, different types of delegation needs to be supported because of the unpredictability of the application.

Researchers have proposed various access control models for pervasive computing applications. Several works exist that focus on how RBAC can be extended to make it context aware [6,5,15]. Other extensions to RBAC include the Temporal Role-Based Access Control Model (TRBAC) [2] and the Generalized Temporal Role Based Access Control Model (GTRBAC) [9]. Researchers have also extended RBAC to incorporate spatial information [3,14]. Incorporating both time and location in RBAC is also addressed by other researchers [1,4,12,16]. Location-based access control has been addressed in other works not pertaining to RBAC [7,8,10,13,11,17]. However, none of these works focus on delegation which is a necessity in pervasive computing applications.

In this paper, we propose a formal access control model for pervasive computing applications. The model extends the one proposed in our earlier work [12]. Since RBAC is policy-neutral, simplifies access management, and widely used by commercial applications, we base our work on it. We show how RBAC can be extended to incorporate environmental contexts, such as time and location. We illustrate how each component of RBAC is related with time and location and show how it is affected by them. We also show how spatio-temporal information is used for making access decisions. We also describe the different types of delegation that are supported by our model. Some of these are constrained by temporal and spatial conditions. The correct behavior of this model is formulated in terms of constraints that must be satisfied by any application using this model.

2 Our Model

Representing Location

There are two types of locations: *physical* and *logical*. All users and objects are associated with locations that correspond to the physical world. These are referred to as the physical locations. A *physical location* $PLoc_i$ is a non-empty set of points $\{p_i, p_j, \ldots, p_n\}$ where a point p_k is represented by three co-ordinates. Physical locations are grouped into symbolic representations that will be used by applications. We refer to these symbolic representations as logical locations. Examples of logical locations are Fort Collins, Colorado etc. A *logical location* is an abstract notion for one or more physical locations. We assume the existence of two translation functions, m and m', that convert from logical locations to physical locations and vice-versa.

Although different kinds of relationships may exist between a pair of locations, here we focus only on *containment* relation. A physical location $ploc_j$ is said to be *contained* in another physical location $ploc_k$, denoted as, $ploc_j \subseteq ploc_k$, if the following condition holds: $\forall p_i \in ploc_j, p_i \in ploc_k$. The location $ploc_j$ is called the contained location and $ploc_k$ is referred to as the containing or the enclosing location. A logical location $lloc_m$ is contained in $lloc_n$, denoted as, $lloc_m \subseteq lloc_n$, if and only if the physical location corresponding to $lloc_m$ is contained within that of $lloc_m$, that is $m'(lloc_m) \subseteq m'(lloc_n)$. We assume the existence of a logical location called *universe* that contains all other locations. In the rest of the paper, we do not discuss physical locations any more. The locations referred to are logical locations.

Representing Time

A *time instant* is one discrete point on the time line. The exact granularity of a time instant will be application dependent. For instance, in some application a time instant may be measured at the nanosecond level and in another one it may be specified at the millisecond level. A *time interval* is a set of time instances. Example of an interval is 9:00 a.m. to 3:00 p.m. on 25th December. Example of another interval is 9:00 a.m. to 6:00 p.m. on Mondays to Fridays in the month of March. We use the notation $t_i \in d$ to mean that t_i is a time instance in the time interval d. A special case of relation between two time intervals that we use is referred to as *containment*. A time interval tv_i is contained in another interval tv_j, denoted as $tv_i \subseteq tv_j$, if the set of time instances in tv_i is a subset of those in tv_j. We introduce a special time interval, which we refer to as *always*, that includes all other time intervals.

Relationship of Core-RBAC entities with Location and Time

We discuss how the different entities of core RBAC, namely, *Users*, *Roles*, *Sessions*, *Permissions*, *Objects* and *Operations*, are associated with location and time.

Users

We assume that each valid user carries a locating device which is able to track his location. The location of a user changes with time. The relation $UserLocation(u,t)$ gives the location of the user at any given time instant t. Since a user can be associated with only one location at any given point of time, we have the following constraint:
$UserLocation(u,t) = l_i \wedge UserLocation(u,t) = l_j \Leftrightarrow (l_i \subseteq l_j) \vee (l_j \subseteq l_i)$
We define a similar function $UserLocations(u,d)$ that gives the location of the user during the time interval d. Note that, a single location can be associated with multiple users at any given point of time.

Objects

Objects can be physical or logical. Example of a physical object is a computer. Files are examples of logical objects. Physical objects have devices that transmit their location information with the timestamp. Logical objects are stored in physical objects. The location and timestamp of a logical object corresponds to the location and time of the physical object containing the logical object. We assume that each object is associated with one location at any given instant of time. Each location can be associated with many objects. The function *ObjLocation(o,t)* takes as input an object o and a time instance t and returns the location associated with the object at time t. Similarly, the function *ObjLocations(o,d)* takes as input an object o and time interval d and returns the location associated with the object.

Roles

We have three types of relations with roles. These are user-role assignment, user-role activation, and permission-role assignment. We begin by focusing on user-role assignment. Often times, the assignment of user to roles is location and time dependent. For instance, a person can be assigned the role of U.S. citizen only in certain designated locations and at certain times only. To get the role of conference attendee, a person must register at the conference location during specific time intervals. Thus, for a user to be assigned a role, he must be in designated locations during specific time intervals. In our model, a user must satisfy spatial and temporal constraints before roles can be assigned. We capture this with the concept of *role allocation*. A role is said to be *allocated* when it satisfies the temporal and spatial constraints needed for role assignment. A role can be assigned once it has been allocated. $RoleAllocLoc(r)$ gives the set of locations where the role can be allocated. $RoleAllocDur(r)$ gives the time interval where the role can be allocated. Some role s can be allocated anywhere, in such cases $RoleAllocLoc(s) = universe$. Similarly, if role p can be assigned at any time, we specify $RoleAllocDur(p) = always$.

Some roles can be activated only if the user is in some specific locations. For instance, the role of audience of a theater can be activated only if the user is in the theater when the show is on. The role of conference attendee can be activated only if the user is in the conference site while the conference is in session. In short, the user must satisfy temporal and location constraints before a role can be activated. We borrow the concept of *role-enabling* [2,9] to describe this. A role is said to be *enabled* if it satisfies the temporal and location constraints needed to activate it. A role can be activated only if it has been enabled. $RoleEnableLoc(r)$ gives the location where role r can be activated and $RoleEnableDur(r)$ gives the time interval when the role can be activated.

The predicate $UserRoleAssign(u,r,d,l)$ states that the user u is assigned to role r during the time interval d and location l. For this predicate to hold, the location of the user when the role was assigned must be in one of the locations where the role allocation can take place. Moreover, the time of role assignment must be in the interval when role allocation can take place.

$$UserRoleAssign(u,r,d,l) \Rightarrow$$
$$(UserLocation(u,d) = l) \wedge (l \subseteq RoleAllocLoc(r)) \wedge (d \subseteq RoleAllocDur(r))$$

The predicate $UserRoleActivate(u,r,d,l)$ is true if the user u activated role r for the interval d at location l. This predicate implies that the location of the user during the role activation must be a subset of the allowable locations for the activated role and all times instances when the role remains activated must belong to the duration when the role can be activated and the role can be activated only if it is assigned.

$$UserRoleActivate(u,r,d,l) \Rightarrow$$
$$(l \subseteq RoleEnableLoc(r)) \wedge (d \subseteq RoleEnableDur(r)) \wedge UserRoleAssign(u,r,d,l)$$

The additional constraints imposed upon the model necessitates changing the preconditions of the functions $AssignRole$ and $ActivateRole$. The permission role assignment is discussed later.

Sessions

In mobile computing or pervasive computing environments, we have different types of sessions that can be initiated by the user. Some of these sessions can be location-dependent, others not. Thus, sessions are classified into different types. Each instance of a session is associated with some type of a session. The type of session instance s is given by the function $Type(s)$. The type of the session determines the allowable location. The allowable location for a session type st is given by the function $SessionLoc(st)$. When a user u wants to create a session si, the location of the user for the entire duration of the session must be contained within the location associated with the session. The predicate $SessionUser(u,s,d)$ indicates that a user u has initiated a session s for duration d.

$SessionUser(u,s,d) \Rightarrow (UserLocation(u,d) \subseteq SessionLoc(Type(s)))$

Since sessions are associated with locations, not all roles can be activated within some session. The predicate $SessionRole(u,r,s,d,l)$ states that user u initiates a session s and activates a role for duration d and at location l.

$SessionRole(u,r,s,d,l) \Rightarrow UserRoleActivate(u,r,d,l) \wedge l \subseteq SessionLoc(Type(s)))$

Permissions

Our model also allows us to model real-world requirements where access decision is contingent upon the time and location associated with the user and the object. For example, a teller may access the bank confidential file if and only if he is in the bank and the file location is the bank secure room and the access is granted only during the working hours. Our model should be capable of expressing such requirements.

Permissions are associated with roles, objects, and operations. We associate three additional entities with permission to deal with spatial and temporal constraints: user location, object location, and time. We define three functions to retrieve the values of these entities. $PermRoleLoc(p,r)$ specifies the allowable locations that a user playing the role r must be in for him to get permission p. $PermObjLoc(p,o)$ specifies the allowable locations that the object o must be in so that the user has permission to operate on the object o. $PermDur(p)$ specifies the allowable time when the permission can be invoked.

We define another predicate which we term *PermRoleAcquire(p,r,d,l)*. This predicate is true if role r has permission p for duration d at location l. Note that, for this predicate to be true, the time interval d must be contained in the duration where the permission can be invoked and the role can be enabled. Similarly, the location l must be contained in the places where the permission can be invoked and role can be enabled.

$$PermRoleAcquire(p,r,d,l) \Rightarrow$$
$$(l \subseteq (PermRoleLoc(p,r) \cap RoleEnableLoc(r))) \wedge (d \subseteq (PermDur(p) \cap RoleEnableDur(p)))$$

The predicate $PermUserAcquire(u,o,p,d,l)$ means that user u can acquire the permission p on object o for duration d at location l. This is possible only when the permission p is assigned some role r which can be activated during d and at location l, the user location and object location match those specified in the permission, the duration d matches that specified in the permission.

$$PermRoleAcquire(p,r,d,l) \wedge UserRoleActivate(u,r,d,l)$$
$$\wedge (ObjectLocation(o,d) \subseteq PermObjectLoc(p,o)) \Rightarrow PermUserAcquire(u,o,p,d,l)$$

For lack of space, we do not discuss the impact of time and location on role-hierarchy or separation of duty, but refer the interested reader to one of our earlier paper [12].

Impact of Time and Location on Delegation

Many situations require the temporary transfer of access rights to accomplish a given task. For example, in a pervasive computing application, a doctor may give certain privilege to a trained nurse, when he is taking a short break. In such situations, the doctor can give a subset of his permissions to the nurse for a given period of time. There are a number of different types of delegation. The entity that transfers his privileges temporarily to another entity is often referred to as the delegator. The entity who receives the privilege is known as the delegatee. The delegator (delegatee) can be either an user or a role. Thus, we may have four types of delegations: *user to user* (U2U), *user to role* (U2R), *role to role* (R2R), and *role to user* (R2U). System administrators are responsible for overseeing delegation when the delegator is a role. Individual users administer delegation when the delegator is an user. When a user is the delegator, he can delegate a subset of permissions that he possesses by virtue of being assigned to different roles. When a role is the delegator, he can delegate either a set of permissions or he can delegate the entire role. We can therefore classify delegation on the basis of role delegation or permission delegation. We identify the following types of delegation.

[U2U Unrestricted Permission Delegation]: In this type of delegation, the delegatee can invoke the delegator's permissions at any time and at any place where the delegator could invoke those permissions. The illness of the company president caused him to delegate his email reading privilege to his secretary.

Let $DelegateU2U\_P_u(u,v,Perm)$ be the predicate that allows user u to delegate the permissions in the set $Perm$ to user v without any temporal or spatial constraints. This will allow v to invoke the permissions at any time or at any place.

$$\forall p \in Perm, DelegateU2U\_P_u(u,v,Perm) \wedge PermUserAcquire(u,o,p,d,l) \Rightarrow PermUserAcquire(v,o,p,d,l)$$

[U2U Time Restricted Permission Delegation]: Here the delegator places time restrictions on when the delegatee can invoke the permissions. However, no special restrictions are placed with respect to location – the delegatee can invoke the permission at any place that the delegator could do so. The professor can delegate his permission to proctor an exam to the teaching assistant while he is on travel.

Let $DelegateU2U\_P_t(u,v,Perm,d')$ be the predicate that allows user u to delegate the permissions in the set $Perm$ to user v for the duration d'.

$$\forall p \in Perm, DelegateU2U\_P_t(u,v,Perm,d') \wedge PermUserAcquire(u,o,p,d,l) \wedge (d' \subseteq d) \Rightarrow PermUserAcquire(v,o,p,d',l)$$

[U2U Location Restricted Permission Delegation]: A delegator can place spatial restrictions on when the delegatee can invoke the permissions. However, the only temporal restriction is that the delegatee can invoke the permissions during the period when the original permission is valid. The teaching assistant can delegate the permission regarding lab supervision to the lab operator only in the Computer Lab.

Let $DelegateU2U\_P_l(u,v,Perm,l')$ be the predicate that allows user u to delegate the permissions in the set $Perm$ to user v in the location l'.

$\forall p \in Perm, DelegateU2U\_P_l(u,v,Perm,l') \wedge PermUserAcquire(u,o,p,d,l) \wedge (l' \subseteq l)$
$\quad \Rightarrow PermUserAcquire(v,o,p,d,l')$

[U2U Time Location Restricted Permission Delegation]: In this case, the delegator imposes a limit on the time and the location where the delegatee can invoke the permission. A nurse can delegate his permission to oversee a patient while he is resting in his room to a relative.

Let $DelegateU2U\_P_{tl}(u,v,Perm,d',l')$ be the predicate that allows user u to delegate the permissions in the set *Perm* to user v in the location l' for the duration d'.
$\forall p \in Perm, DelegateU2U\_P_{tl}(u,v,Perm,t',l') \wedge PermUserAcquire(u,o,p,d,l)$
$\quad \wedge (d' \subseteq d) \wedge (l' \subseteq l) \Rightarrow PermUserAcquire(v,o,p,d',l')$

[U2U Unrestricted Role Delegation]: Here the delegator delegates a role to the delegatee. The delegatee can activate the roles at any time and place where the delegator can activate those roles. A manager before relocating can delegate his roles to his successor in order to train him.

Let $DelegateU2U\_R_u(u,v,r)$ be the predicate that allows user u to delegate his role r to user v.
$DelegateU2U\_R_u(u,v,r) \wedge UserRoleActivate(u,r,d,l) \Rightarrow UserRoleActivate(v,r,d,l)$

[U2U Time Restricted Role Delegation]: In this case, the delegator delegates a role to the delegatee but the role can be activated only for a more limited duration than the original role. A user can delegate his role as a teacher to a responsible student while he is in a conference.

Let $DelegateU2U\_R_t(u,v,r,d')$ be the predicate that allows user u to delegate his role r to user v for the duration d'.
$DelegateU2U\_R_t(u,v,r,d') \wedge UserRoleActivate(u,r,d,l) \wedge$
$\quad (d' \subseteq RoleEnableDur(r)) \wedge (d' \subseteq d) \Rightarrow UserRoleActivate(v,r,d',l)$

[U2U Location Restricted Role Delegation]: In this case, the delegator delegates a role to the delegatee but the role can be activated in more limited locations than the original role. A student can delegate his lab supervision role to another student in a designated portion of the lab only.

Let $DelegateU2U\_R_l(u,v,r,l')$ be the predicate that allows user u to delegate his role r to user v in the location l'.
$Delegate\_R_l(u,v,r,l') \wedge UserRoleActivate(u,r,d,l) \wedge$
$\quad (l' \subseteq RoleEnableLoc(r)) \wedge (l' \subseteq l) \Rightarrow UserRoleActivate(v,r,d,l')$

[U2U Time Location Restricted Role Delegation]: The delegator delegates the role, but the delegatee can activate the role for a limited duration in limited places. A student can delegate his lab supervision role to another student only in the lab when he leaves the lab for emergency reasons.

Let $DelegateU2U\_R_{tl}(u,v,r,d',l')$ be the predicate that allows user u to delegate his role r to user v in location l' and time d'.
$DelegateU2U\_R_{tl}(u,v,r,d',l') \wedge UserRoleActivate(u,r,d,l) \wedge (l' \subseteq RoleEnableLoc(r)) \wedge$
$\quad (d' \subseteq RoleEnableDur(r)) \wedge (d' \subseteq d) \wedge (l' \subseteq l) \Rightarrow UserRoleActivate(v,r,d',l')$

[R2R Unrestricted Permission Delegation]: Here, all users assigned to the delegatee role can invoke the delegator role's permissions at any time and at any place where the user of this delegator role could invoke those permissions. The Smart Home owner role may delegate the permission to check the status of security sensors of the home to the police officer role, so all police officers can detect the intruder at any time at any place.

Let $DelegateR2R\_P_u(r_1, r_2, Perm)$ be the predicate that allows role r_1 to delegate the permissions in the set $Perm$ to role r_2 without any temporal or spatial constraints. This will allow users in the role r_2 to invoke the permissions at any time or at any place.

$$\begin{aligned}&\forall p \in Perm, DelegateR2R\_P_u(r_1, r_2, Perm) \wedge PermRoleAcquire(p, r_1, d, l) \wedge\\ &\quad (d \subseteq RoleEnableDur(r_2)) \wedge (l \subseteq RoleEnableLoc(r_2))\\ &\quad \Rightarrow PermRoleAcquire(p, r_2, d, l)\end{aligned}$$

[R2R Time Restricted Permission Delegation]: The delegator role can place temporal restrictions on when the users of the delegatee role can invoke the permissions. No special restrictions are placed with respect to location i.e. the delegatee role's users can invoke the permissions at any place that the delegator role's users could do so. CS599 teacher role can grant the permission to access course materials to CS599 student role for the specific semester.

Let $DelegateR2R\_P_t(r_1, r_2, Perm, d')$ be the predicate that allows role r_1 to delegate the permissions in the set $Perm$ to role r_2 for the duration d'.

$$\begin{aligned}&\forall p \in Perm, DelegateR2R\_P_t(r_1, r_2, Perm, d') \wedge (d' \subseteq d)\ PermRoleAcquire(p, r_1, d, l) \wedge\\ &\quad (l' \subseteq l) \wedge (d' \subseteq RoleEnableDur(r_2)) \wedge (l \subseteq RoleEnableLoc(r_2))\\ &\quad \Rightarrow PermRoleAcquire(p, r_2, d', l)\end{aligned}$$

[R2R Location Restricted Permission Delegation]: Here, the delegator role places spatial constraints on where the users of the delegatee role can invoke the permissions. No special temporal constraints are placed, that is, the delegatee role's users can invoke the permissions at any time that the delegator role's users could do so. The librarian role may grant the permission to checkout the book to the student role only at the self-checkout station.

Let $DelegateR2R\_P_l(r_1, r_2, Perm, l')$ be the predicate that allows role r_1 to delegate the permissions in the set $Perm$ to role r_2 in the location l'.

$$\begin{aligned}&\forall p \in Perm, DelegateR2R\_P_l(r_1, r_2, Perm, l') \wedge PermRoleAcquire(p, r_1, d, l) \wedge\\ &\quad (d \subseteq RoleEnableDur(r_2)) \wedge (l' \subseteq RoleEnableLoc(r_2)) \wedge (l' \subseteq l)\\ &\quad \Rightarrow PermRoleAcquire(p, r_2, d, l')\end{aligned}$$

[R2R Time Location Restricted Permission Delegation]: Here the delegator role imposes a limit on the time and the location where the delegatee role's users could invoke the permissions. The daytime doctor role may delegate the permission to get his location information to the nurse role only when he is in the hospital during the daytime.

Let $DelegateR2R\_P_{tl}(r_1, r_2, Perm, d', l')$ be the predicate that allows role r_1 to delegate the permissions in the set $Perm$ to role r_2 in the location l' for the duration d'.

$$\begin{aligned}&\forall p \in Perm, DelegateR2R\_P_{tl}(r_1, r_2, Perm, d', l') \wedge PermRoleAcquire(p, r_1, d, l) \wedge\\ &\quad (d' \subseteq RoleEnableDur(r_2)) \wedge (l' \subseteq RoleEnableLoc(r_2)) \wedge (d' \subseteq d) \wedge (l' \subseteq l)\\ &\quad \Rightarrow PermRoleAcquire(p, r_2, d', l')\end{aligned}$$

[R2R Unrestricted Role Delegation]: Here all users assigned to the delegatee role can activate the delegator role at any time and at any place where the user of this delegator role could activate the role. In the case of company reorganization, the manager role can be delegated to the manager successor role in order to train him.

Let $DelegateR2R\_R_u(r_1, r_2)$ be the predicate that allows role r_1 to be delegated to role r_2.

$$DelegateR2R\_R_u(r_1, r_2) \wedge UserRoleActivate(u, r_2, d, l) \wedge (d \subseteq RoleEnableDur(r_1)) \wedge (l \subseteq RoleEnableLoc(r_1)) \Rightarrow UserRoleActivate(u, r_1, d, l)$$

[R2R Time Restricted Role Delegation]: Here, the delegator places temporal constraints on when the users of the delegatee role can activate the delegator role. No special spatial constraints are placed i.e. the delegatee role's users can activate the delegator role at any place that the delegator role's users could do so. The permanent staff role can be delegated to the contract staff role during the contract period.

Let $DelegateR2R\_R_t(r_1, r_2, d')$ be the predicate that allows role r_1 to be delegated to role r_2 for the duration d'.

$$DelegateR2R\_R_t(r_1, r_2, d') \wedge UserRoleActivate(u, r_2, d', l) \wedge (d \subseteq RoleEnableDur(r_1)) \wedge (l \subseteq RoleEnableLoc(r_1)) \wedge (d' \subseteq d) \Rightarrow UserRoleActivate(u, r_1, d', l)$$

[R2R Location Restricted Role Delegation]: The delegator role can place spatial restrictions on where the users of the delegatee role can activate the delegator role. No special restrictions are placed with respect to time i.e. the delegatee role's users can activate the delegator role at any time that the delegator role's users could do so. The researcher role can be delegated to the lab assistant role at the specific area of the lab.

Let $DelegateR2R\_R_l(r_1, r_2, l')$ be the predicate that allows role r_1 to be delegated to role r_2 in the location l'.

$$DelegateR2R\_R_l(r_1, r_2, l') \wedge UserRoleActivate(u, r_2, d, l') \wedge (d \subseteq RoleEnableDur(r_1)) \wedge (l \subseteq RoleEnableLoc(r_1)) \wedge (l' \subseteq l) \Rightarrow UserRoleActivate(u, r_1, d, l')$$

[R2R Time Location Restricted Role Delegation]: In this case, the delegator role imposes a limit on the time and the location where the delegatee role's users could activate the role. The full-time researcher role can be delegated to the part-time researcher role only during the hiring period in the specific lab.

Let $DelegateR2R\_R_{tl}(r_1, r_2, d', l')$ be the predicate that allows role r_1 to be delegated to role r_2 in the location l' for the duration d'.

$$DelegateR2R\_R_{tl}(r_1, r_2, d', l') \wedge UserRoleActivate(u, r_2, d', l') \wedge (d' \subseteq d) \wedge (l' \subseteq l) \wedge (d \subseteq RoleEnableDur(r_1)) \wedge (l \subseteq RoleEnableLoc(r_1)) \wedge (d' \subseteq d) \wedge (l' \subseteq l) \Rightarrow UserRoleActivate(u, r_1, d', l')$$

3 Conclusion and Future Work

Traditional access control models which do not take into account environmental factors before making access decisions may not be suitable for pervasive computing applications. Towards this end, we proposed a spatio-temporal role based access control model that supports delegation. The behavior of the model is formalized using constraints.

An important work that we plan to do is the analysis of the model. We have proposed many different constraints. We are interested in finding conflicts and redundancies among the constraint specification. Such analysis is needed before our model can be used for real world applications. We plan to investigate how to automate this analysis. We also plan to implement our model. We need to investigate how to store location and temporal information and how to automatically detect role allocation and enabling using triggers.

References

1. Atluri, V., Chun, S.A.: A geotemporal role-based authorisation system. International Journal of Information and Computer Security 1(1/2), 143–168 (2007)
2. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A Temporal Role-Based Access Control Model. In: Proceedings of the 5th ACM workshop on Role-Based Access Control, Berlin, Germany, July 2000, pp. 21–30. ACM Press, New York (2000)
3. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: a spatially aware RBAC. In: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, Sweden, June 2005, pp. 29–37. ACM Press, New York (2005)
4. Chandran, S.M., Joshi, J.B.D.: LoT-RBAC: A Location and Time-Based RBAC Model. In: WISE, pp. 361–375 (2005)
5. Covington, M.J., Fogla, P., Zhan, Z., Ahamad, M.: A Context-Aware Security Architecture for Emerging Applications. In: Proceedings of the Annual Computer Security Applications Conference, Las Vegas, NV, USA, December 2002, pp. 249–260 (2002)
6. Covington, M.J., Long, W., Srinivasan, S., Dey, A., Ahamad, M., Abowd, G.: Securing Context-Aware Applications Using Environment Roles. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies, Chantilly, VA, USA, May 2001, pp. 10–20 (2001)
7. Hengartner, U., Steenkiste, P.: Implementing Access Control to People Location Information. In: Proceeding of the 9th Symposium on Access Control Models and Technologies, Yorktown Heights, New York (June 2004)
8. Hulsebosch, R.J., Salden, A.H., Bargh, M.S., Ebben, P.W.G., Reitsma, J.: Context sensitive access control. In: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, Sweden, pp. 111–119. ACM Press, New York (2005)
9. Joshi, J.B.D., Bertino, E., Latif, U., Ghafoor, A.: A Generalized Temporal Role-Based Access Control Model. IEEE Transactions on Knowledge and Data Engineering 17(1), 4–23 (2005)
10. Leonhardt, U., Magee, J.: Security Consideration for a Distributed Location Service. Imperial College of Science, Technology and Medicine, London, UK (1997)
11. Pu, F., Sun, D., Cao, Q., Cai, H., Yang, F.: Pervasive Computing Context Access Control Based on $UCON_{ABC}$ Model. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP 2006, December 2006, pp. 689–692 (2006)
12. Ray, I., Toahchoodee, M.: A Spatio-Temporal Role-Based Access Control Model. In: Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, July 2007, pp. 211–226 (2007)
13. Ray, I., Kumar, M.: Towards a Location-Based Mandatory Access Control Model. Computers & Security 25(1) (February 2006)
14. Ray, I., Kumar, M., Yu, L.: LRBAC: A Location-Aware Role-Based Access Control Model. In: Proceedings of the 2nd International Conference on Information Systems Security, Kolkata, India, December 2006, pp. 147–161 (2006)

15. Sampemane, G., Naldurg, P., Campbell, R.H.: Access Control for Active Spaces. In: Proceedings of the Annual Computer Security Applications Conference, Las Vegas, NV, USA, December 2002, pp. 343–352 (2002)
16. Samuel, A., Ghafoor, A., Bertino, E.: A Framework for Specification and Verification of Generalized Spatio-Temporal Role Based Access Control Model. Technical report, Purdue University, February 2007. CERIAS TR 2007-08 (2007)
17. Yu, H., Lim, E.-P.: LTAM: A Location-Temporal Authorization Model. In: Secure Data Management, pp. 172–186 (2004)

A Mechanism for Ensuring the Validity and Accuracy of the Billing Services in IP Telephony

Dimitris Geneiatakis, Georgios Kambourakis, and Costas Lambrinoudakis

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece
{dgen,gkamb,clam}@aegean.gr

Abstract. The current penetration, but also the huge potential, of Voice over IP (VoIP) telephony services in the market, boosts the competition among telecommunication service providers who promote new services through many different types of offers. However, this transition from the closed Public Switched Telephone Network (PSTN) architecture to the internet based VoIP services, has resulted in the introduction of several threats both intrinsic i.e. VoIP specific, and Internet oriented. In the framework of this paper, we are considering threats that may affect the accuracy and validity of the records of the billing system that the service provider is using for charging the users. We are proposing a simple, practical and effective mechanism for protecting telecommunication service providers and end users from malicious activities originated from the end users and telecommunication service providers respectively. In both cases the malicious activity concerns fraud through the billing system. The proposed mechanism focuses on VoIP services that are based on the Session Initiation Protocol (SIP). However, it can be easily amended to cover other VoIP signaling protocols, as it takes advantage of the underlying AAA network infrastructure to deliver robust time stamping services to SIP network entities.

Keywords: Session Initiation Protocol (SIP), Billing, Voice Over IP (VoIP).

1 Introduction

The advent of Voice over IP (VoIP) Telephony[1] services offers to Telecommunication Service Providers (TSPs) new opportunities for providing advanced services, like conference rooms, click to dial, and multimedia delivery. In PSTN such services could not be realized at a large scale and at a relatively low cost. Furthermore, the potential of such services is highlighted by the estimation that up to the year 2012, VoIP users would reach the number of twelve million. Note, that currently the number VoIP users is not more that one million [1]. However, in order for TSPs to support such services, they should, among other things, provide accurate accounting services and particularly billing. This will boost the trustworthiness and popularity of VoIP services to potential consumers and will greatly increase IP telephony market share.

[1] Hereafter the terms Voice over IP and IP Telephony services are considered equivalent.

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 59–68, 2008.

Several researchers [2]-[4] have already identified various types of attacks that could be launched against VoIP services. Such attacks can severely affect not only the end-users but also the VoIP providers and the underlying infrastructure. Putting aside Quality of Service (QoS) issues, when end-users acquire VoIP services they are mostly worried about the accuracy and validity of their billing accounts. For example, the service provider could act maliciously and modify in an illegal way the Call Detail Records (CDRs) in order to overcharge the billing account of a given end-user. In the same way, an end-user could repudiate the calls included in his billing account in order to avoid the charges. It should be stressed that in such cases neither the end-user nor the TSP can prove the validity of the CDRs due to the lack of the appropriate non-repudiation mechanisms in IP Telephony services.

This paper proposes a simple, practical and effective mechanism for protecting, both end-users and TSPs, from billing frauds. While our mechanism focuses on VoIP services that are based on Session Initiation Protocol (SIP) [5], it can be easily amended to cover other VoIP signaling protocols as well. This is because our scheme takes advantage of the underlying AAA network infrastructure to deliver robust time stamping services to SIP network entities.

The rest of the paper is organized as follows. Section 2 provides background information regarding billing services in VoIP. Section 3 presents various security incidents that affect the validity and accuracy of the billing service, while Section 4 introduces and thoroughly analyzes the proposed scheme. Finally, Section 5 concludes the paper giving directions for future work.

2 Billing Services in IP Telephony

VoIP attracts gradually more and more subscribers [1] and as already mentioned it is anticipated to gain a significant market share in the next few years. This growth is actually driven by two key factors: the low cost of VoIP service acquisition and the similar levels of QoS when compared to those of PSTN. TSPs do promote VoIP services through various offers, like free usage time, lower costs for prepaid cards and many more. In fact, all these offers are based on different billing methods that the TSPs must support. According to [6],[7] the billing methods that are available for services provided through the Internet architecture can be classified into the following categories:

- *Fixed Charge*: The subscriber pays a fixed subscription fee for a specific period of time (e.g. monthly) irrespectively of the service usage.
- *Usage Charge*: The subscriber pays a fee based on service usage (e.g. the volume of the data being transferred). For Internet oriented services two basic usage models are employed: (a) Service Time usage and (b) Transferred Media. Note that the latter model is not suitable for voice services.
- *Service Quality Charge*: According to this model whenever the subscriber access the service, he pays a fee based on the provided QoS offered by the TSP as the case may be.

Nowadays most TSPs employ mixed billing schemes, combining *Fixed* and *Usage* charging schemes, which rely either on prepaid or post billing services. However, in every case the employed billing scheme does not influence the general accounting method in use. To be more precise, by the term *accounting method* we refer to the process of collecting information about chargeable events, which will be later used as input to the billing service. Consequently, the billing process for IP telephony requires, among others, accurate tracing of "start" and "end" events for all the services acquired by a given subscriber in order to charge him appropriately. These events are known as *Call Detail Records* (CDRs). An example of such an event logging process sequence in a SIP based VoIP service, is presented in Table 1. Normally, CDRs are captured either by the Authentication, Authorization and Accounting (AAA) Server in charge, or the corresponding SIP proxy, depending on the service configuration parameters.

Table 1. An Example of Call Detail Records

Call-Id	Caller	Callee	Type Msg	Time-Date
123@sip.gr	dgen@sip.gr	gkar@sip.gr	INVITE	1/1/2008:11:00:00
123@sip.gr	dgen@sip.gr	gkar@sip.gr	200 OK	1/1/2008:11:00:01
123@sip.gr	dgen@sip.gr	gkar@sip.gr	BYE	1/1/2008:11:05:04

Let us consider a User A (caller) who wishes to establish a voice connection with a User B (callee), through some specific SIP based VoIP service. First of all, the caller generates a SIP INVITE message and sends it to the corresponding SIP proxy, which in turn forwards it to the callee. It is assumed that the caller must have been previously authenticated by the local AAA server which is responsible to authorize

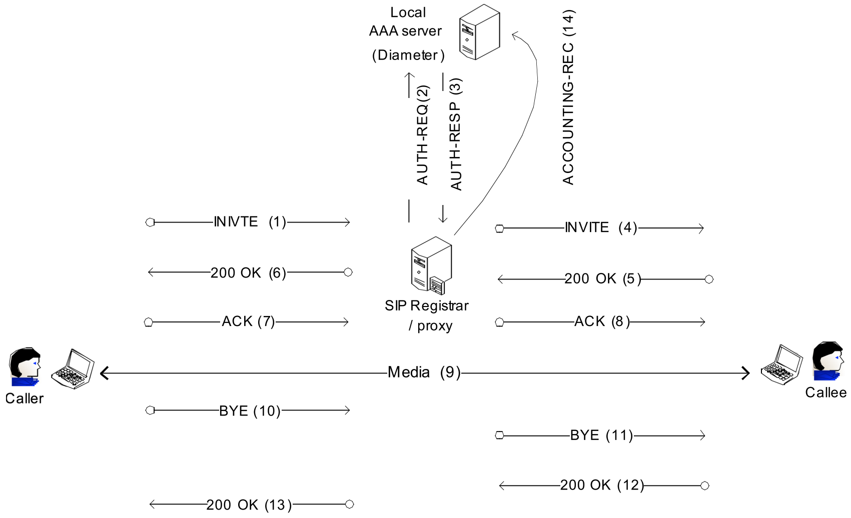


Fig. 1. Call Establishment procedure in SIP based IP Telephony Services

him (or not) to access the voice service. Other interactions are also possible in this stage, i.e. if the user is roaming to a foreign domain the local AAA server may contact the AAA server of the caller's home domain in order to obtain the proper authentication and/or authorization credentials. However, for the sake of simplicity of the example, we assume that the caller uses a postpaid service. Provided that the callee is available, the session will be successfully established after the caller sends, through the SIP proxy, the final ACK message to the callee. Whenever any of the participants wishes to terminate the session, he issues a BYE message. Upon that, the SIP proxy is responsible to send to the AAA server (immediately or at a latter time) the corresponding CDRs that will be used as input to the billing service. The aforementioned procedure is depicted in Figure 1.

3 Billing Attacks against IP Telephony

Fraud attempts could be launched against any telecommunication system by employing several different methods and techniques. According to [8]there are 200 types of known telecommunication frauds. However, the closed architecture of PSTN offers very few opportunities to malicious users for frauds through the manipulation of signaling data. A well known fraud incident in PSTN took place in early 1960's, when common associated signaling was used by TSPs [9] This attack unfolds as follows: the aggressor sends a termination tone to the call center without hanging on his device. Although the call was terminated successfully, resources related with the previous connection remain allocated since the call center is waiting for the on hook condition. At the same time the malicious user could dial a new telephone number along with a start tone and establish a new connection without charging his account. Currently, the introduction of Common Channel Signaling (CCS), in conjunction with PSTN's closed architecture, makes such type of attacks impossible.

On the contrary, the advent of VoIP which relies on the Internet, introduces several threats both intrinsic i.e. VoIP specific, and Internet oriented. For example, a malevolent user may try to evade charging, or even worse, charge another innocent legitimate user with calls that he has never performed. This is due to the fact that there are several methods that a malicious user could exploit in order to manipulate VoIP signaling data as demonstrated in [3]. Considering the call establishment procedure of Figure 1, a malicious caller instead of sending an ACK message after receiving the "200 OK" response from the callee, manipulates his telephone to suppress it. As a result, the SIP proxy assumes that the call has not been established, but the caller is actually able to communicate with the callee. In another scenario depicted in Figure 2, a malicious user may act as a *Man In The Middle* (MITM) in order to modify an INVITE message. That is, the INVITE's message *Contact* header is set to the malicious user IP address and the *To* header to that of the person that the malicious user wishes to communicate with. The spoofed INVITE is then forwarded towards the corresponding proxy. The proxy sends the request towards the callee who, after accepting the call, generates a "200 OK" response message which is finally passed to the malicious user. Upon receiving it, the attacker replaces it with a "Busy" message and forwards it to the legitimate user who acknowledges the spoofed Busy response

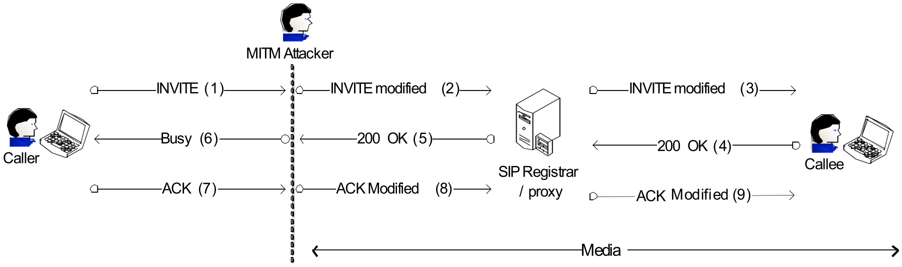


Fig. 2. An Example of Man-In-The-Middle Attack during Call Establishment

and terminates the session. Under this context, the conversation between the malicious user and the callee has been successfully established, while the malicious user wangled to debit a legitimate user's account for a call that did not actually made.

Similar techniques are used in billing attacks known as *"Fake busy"*, *"Bye Delay"* and *"ByeDrop"*, which are discussed in detail in [10]. The difference between the aforementioned scenario and the *"Fake Busy"* one is the existence of another malicious user acting on behalf of the callee. This second aggressor intercepts the SIP messages and generates a "200 OK" response in order to make his IP address available to his collaborator in the attack which is placed on the side of the caller. After that, a media session between the two attackers can be successfully established. In this case the (legitimate) callee is unaware of the incoming invitation. As far as the rest of the attack scenarios, i.e. *Bye Delay* and *Bye-Drop*, the MITM attacker captures the SIP BYE message that the legitimate user (caller or callee) sends to the Proxy and sends back to the (legitimate) user a spoofed "200 OK" response message. This fact gives the impression to the service provider that the session is still active, whereas the legitimate user thinks that the session has been successfully terminated. It should be stated that in all the above security incidents the malicious user attempts to debit the legitimate user for calls that he never made.

4 The Proposed Mechanism

Billing accuracy severely affects end-users' trust to VoIP services. Thus, service providers should employ robust solutions and countermeasures against threats similar to those described in Section 3. Normally, TSPs start charging a caller as soon as the 200 OK response message has been received by the corresponding SIP proxy. This is crucial in order to thwart clever attackers from establishing free calls. Even this countermeasure, however, is not an accurate indication that the session between the two ends has been established successfully. For example, referring to Figure 1, the caller's network may become inoperable before the caller sends the final ACK (message #7). Even though this will interrupt the session in an abnormal way, the TSP will wrongly charge the caller for some service time. It is therefore clear that the employment of mechanisms for protecting both end-users and TSPs against billing frauds is necessary.

4.1 Architecture and General Description

The objective is to introduce a lightweight, practical and effective solution for preserving non-repudiation and non-usurpation in SIP. Thus, the choice was not to use mechanisms that mandate Public Key Infrastructure (PKI), like [12]. For the same reason we have set aside recent solutions [13]that are more generic and require third network entities or additional components. The proposed scheme is fully compatible with the underlying network infrastructure and the protocols employed. Moreover, it can support roaming users and heterogeneous network realms towards 4G. Figure 3 depicts the general architecture and the message flow of the proposed scheme. A detailed description of each step follows.

1. At first, the SIP user authenticates himself to the network using a standard AAA protocol. Here, we select Diameter [14], but RADIUS [16] is also an option without any loss of generality. Note, that up to this point, the user authentication is performed by utilizing any sort of credentials that the user has, via standard EAP methods, like EAP-TLS [17], EAP-AKA [18], etc.
2. Secondly, the user needs to register with the SIP registrar server in order to be able to receive and make calls. Here, a standard SIP authentication method, like the digest one [19] may be used.
3. After that, when the user initiates a SIP call, the User Agent (UA) sends a standard Diameter accounting request to the local AAA server. It is worth noting that the Accounting-Record-Type Attribute Value Pair (AVP), i.e. AVP Code 480, which contains the type of accounting record being sent, must set to EVENT_RECORD. This value indicates that a one-time event has occurred [14].
4. The AAA server sends a triplet of {Origin host || Session_ID || Timestamp} information to the local SIP proxy. It also keeps a signed, with his private key, copy of the triplet to a log file that could be used in case of dispute. The origin host field contains the IP address of the user's device. The IP address of the local SIP proxy may be pre-configured to the AAA server. If not, there are at least two more ways to find it. Normally, the location information can be discovered dynamically, based on Dynamic Host Control Protocol (DHCP). The DHCP server shall inform the AAA with the domain name of the local SIP proxy and the address of a Domain Name Server (DNS) that is capable to resolve the Fully Qualified Name (FQDN) of the SIP proxy by using DHCP. A second option is to include the IP address of the local SIP proxy to the Diameter accounting request (see step 3).
5. The proxy acknowledges the message, otherwise the AAA server may retransmit it after a given time interval, following SIP's retransmission time settings for a Non-INVITE request [5]. It also stores the received triplet to the corresponding queue. As discussed in the next subsection, for some predetermined time interval the AAA server will ignore any similar requests that originate from the same UA and have the same session_ID.
6. The AAA server responds back to the originating UA with a Diameter accounting response, which contains an Event-Timestamp AVP. As described in [14]a Timestamp AVP, records the time that the reported event occurred. The SIP INVITE procedure begins at this point and assuming that the callee accepts the call, a 200 OK message is returned to the caller.

7. At this point, the UA is ready to start the call by sending a SIP ACK message to the local SIP proxy. Before doing so, the UA concatenates the received timestamp with the SIP ACK message. Upon reception, the SIP proxy will check its queue for a match, i.e. a same {Origin host || Session_ID || Timestamp}. It is noted that the corresponding queue has a limited length. That is, a triplet should remain in the queue until the session exceeds as it is specified in the SIP's Finite State Machine (FSM) [5]. If the matching procedure returns true, the proxy forwards the INVITE message to the caller, probably via other proxies, and logs the event along with the corresponding ACK message. The log files may be collected in batches at a later time by the underlying accounting service.

The same procedure should be followed before the call is terminated, that is, before the corresponding SIP BYE message, to timestamp the event of call termination. Eventually, the start and stop instances of user charging are designated by the two timestamps acquired by the AAA server.

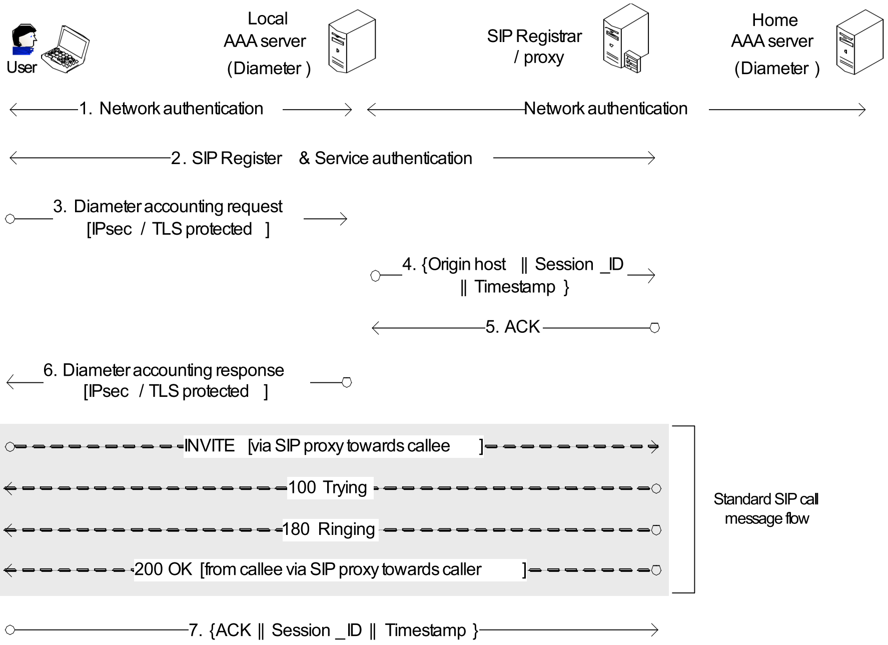

User
Local
AAA server
(Diameter)
SIP Registrar
/ proxy
Home
AAA server
(Diameter)
1. Network authentication
Network authentication
2. SIP Register & Service authentication
3. Diameter accounting request
[IPsec / TLS protected]
4. {Origin host || Session _ID
|| Timestamp }
5. ACK
6. Diameter accounting response
[IPsec / TLS protected]
INVITE [via SIP proxy towards callee]
100 Trying
180 Ringing
200 OK [from callee via SIP proxy towards caller]
Standard SIP call
message flow
7. {ACK || Session _ID || Timestamp }

Fig. 3. Generic architecture and scheme's message flow

4.2 Security Analysis

In terms of security there are several aspects of the proposed scheme that must be carefully examined. The network authentication and the SIP register / authentication phases depend on the authentication methods and security policies employed. However, this is outside the scope of this paper. In fact, our security analysis concentrates on steps 3 to 7. As highlighted in [14] the Diameter protocol must not be used without any security mechanism (TLS or IPsec). Therefore, the communication links between any Diameter nodes (Client, Agent or Server) are considered secure. Furthermore, when end-to-end security is required the End-to-End security extension, known as CMS Security Application [15], may be utilized. As a result, messages 3 & 6 in Figure 3 are considered to be secure when in transit. Nevertheless, an attacker may exploit the Diameter accounting request message to trigger a Denial of Service (DoS) attack against the local AAA server. Such a scenario will possibly enable the attacker to flood the AAA server with Diameter accounting request messages. However, this attack cannot be mounted since, as already mentioned, the AAA server will drop all subsequent requests (arriving after the first one) that originate from the same UA and have the same Session_ID, for a predetermined time interval (see step 5 in the previous subsection). Under these circumstances, the attacker will need a large number of zombies to launch such an attack, having each zombie sending a request every 30 seconds, a scenario that is considered highly improbable. IP spoofing by a limited number of machines is also out of question since all modern routers will easily detect such an attack. Moreover, giving the fact that the {Origin host || Session_ID || Timestamp} queue holds only a limited number of records, overflow style DoS attacks are not feasible.

Another scenario could be the eavesdropper to acquire a Diameter accounting response in order to use it for his own benefit or to just cause commotion to the accounting system. This is however infeasible since the communication between the UA and the AAA server is encrypted and also because the SIP server will match each INVITE SIP message with its own records. Furthermore, in order to protect the integrity and authenticity of SIP ACK and BYE messages, against MITM attacks, a mechanism like the *Interity-Auth* header proposed in [20] should be adopted.

4.3 Resolution of Disputes

Let us now consider a case where a legitimate user repudiates a specific call (or part of it) that has been included in his billing account. If that happens, the TSP will requests from the AAA server the log file of the signed timestamps that correspond to the sessions-calls made. Furthermore, the TSP locates in the SIP proxy logs, the SIP ACK and the corresponding SIP BYE message, designating the start and end of the specific call. With the AAA signed triplet {Origin host || Session_ID || Timestamp} and the user's SIP ACK and BYE messages, the TSP is able to prove that a call was indeed generated by the claimant. The TSP is also able to prove the exact duration of the call. Note that due to the employment of the *Integrity-Auth* scheme [20], only properly authenticated entities can establish or terminate calls by generating the corresponding SIP messages. This ensures that no legitimate user is able to put calls on behalf of another. The claimant may also contend that the TSP generated these messages by his own, relied on the fact that the *Integrity-Auth* scheme is based on a pre-shared password. However, this is not feasible since the AAA (which has the role of a trusted third party) issues timestamps only for requests received by end-users. So, even in cases where the TSP tries to illegally modify a timestamp, he will not be able to match it later with the original AAA's signed timestamp. This means that the user would be able to prove that the corresponding accounting data were illegaly modified.

5 Conclusions and Future Work

Billing mechanisms are of major importance for real-time services, like VoIP. This work elaborates on the accounting process, proposing a novel and robust billing system. The requirements of the proposed mechanism are defined and all the accounting scenarios that the system should cope with are examined. The proposed mechanism is generic and capitalizes on the existing AAA infrastructure, thus providing secure means to transfer and store sensitive billing data. More importantly, it can be easily incorporated into the TSP's existing mechanisms regardless of the underlying network technology. At the same time its generic nature allows for interoperability between different network operators and service providers. The next steps of this work include the implementation and evaluation of a prototype system.

References

[1] VoIP IP Telephony blog (2007), http://snapvoip.blogspot.com/2007/03/virtual-voip-carriers-vvcs-will-grow-to.html

[2] VOIPSA, VoIP Security and Privacy Threat Taxonomy (October 2005), http://www.voipsa.org/Activities/taxonomy.php

[3] Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, K.S., Sisalem, D.: Survey of security vulnerabilities in session initiation protocol. Communications Surveys & Tutorials, IEEE 8(3), 68–81 (2006)

[4] Sisalem, D., Kuthan, J., Ehlert, S.: Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms. Network, IEEE 20(5), 26–31 (2006)

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Spark, R., Handley, M., Schooler, E.: Session Initiation Protocol, RFC 3261 (June 2002)

[6] Wook, H., Han, J., Huh, M., Park, S., Kang, S.: Study, on robust billing mechanism for SIP-based internet telephony services. In: ICACT 2004, vol. 2, pp. 756–759 (2004)

[7] Ibrahim, H.A., Nossier, B.M., Darwish, M.G.: Billing system for Internet service provider (ISP). In: MELECON 2002, pp. 260a–268a (2002)

[8] Thermos, P., Takanen, A.: Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures. Addison-Wesley Professional, Reading (August 2007)

[9] John, G., van Bosse, F.U.: Signaling in Telecommunication Networks, 2nd edn. Wiley InterScience, Chichester (December 2006)

[10] Ruishan, Wang, X., Yang, X., Jiang, X.: Billing Attacks on SIP-Based VoIP Systems. In: Proc. of first USENIX workshop on offensive technologies (August 2007)

[11] Rigney, C.: RADIUS Accounting, RFC 2139 (April 1997)

[12] Adams, C., Cain, P., Pinkas, D., Zuccherato, R.: Entrust Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP), RFC 3161 (August 2001)

[13] Hakala, H., Mattila, L., Koskinen, J.-P., Stura, M., Loughney, J.: Diameter Credit-Control Application, RFC 4006 (August 2005)

[14] Calhoun, P., Loughney, J., Guttman, B., Zorn, G., Arkko, J.: Diameter Base Protocol, IETF RFC 3588 (Septeber 2003)

[15] Calhoun, P., Bulley, W., Farrell, S.: Diameter CMS Security Application (July 2001)

[16] Rigney, C., et al.: Remote Authentication Dial In User Service, RFC 2865 (June 2000)

[17] Aboba, B.B, Simon, D.: PPP EAP-TLS Authentication Protocol, RFC 2716 (October 1999)

[18] Arkko, J., Haverinen, H.: EAP-AKA Authentication, RFC 4187 (January 2006)

[19] Franks, J., et al.: HTTP Authentication: Basic and Digest Access Authentication, RFC 2617 (June 1999)

[20] Geneiatakis, D., Lambrinoudakis, C.: A Lightweight Protection Mechanism against signaling Attacks in a SIP-Based VoIP environment. Telecommunication Systems. Springer, Heidelberg (2008)

Multilateral Secure Cross-Community Reputation Systems for Internet Communities

Franziska Pingel and Sandra Steinbrecher

Technische Universität Dresden, Fakultät Informatik, D-01062 Dresden, Germany

Abstract. The Internet gives people various possibilities to interact with each other. Many interactions need trust that interactors behave in a way one expects them to do. If people are able to build reputation about their past behaviour this might help others to estimate their future behaviour. Reputation systems were designed to store and manage these reputations in a technically efficient way. Most reputation systems were designed for the use in single Internet communities although there are similarities between communities. In this paper we present a multilateral secure reputation system that allows to collect and use reputation in a set of communities interoperable with the reputation system. We implemented our system for the community software phpBB[1].

1 Introduction

Internet communities cover various fields of interest for many people, e.g. marketplaces like eBay[2] or online games like Second Life[3]. Social networks like Facebook[4] contain various communities linked with each other on one platform.

When interacting with others in a community security requirements and trust issues become important. An interactor first wants to know what to expect from others and then wants to trust in the fulfilment of his expectations. Usually only users who fulfil these expectations are seen as trustworthy in the future.

On the Internet users often only interact once with each other. To help new interactors to estimate the others' behaviour reputation systems have been designed and established to collect the experiences former interactors made [11]. A very-popular example of a reputation system is implemented by eBay. As marketplace it offers its members the possibility to sell and buy arbitrary objects. The exchange of object and money 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 collects the experiences sellers and buyers make. After every exchange they may give comments or/and marks to each other that are added to the members' public reputation (usually together with the annotator and the exchange considered as context information).

[1] http://www.phpBB.com

[2] http://www.ebay.com/

[3] http://www.secondlife.com/

[4] http://www.facebook.com/

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 69–78, 2008.

Many people are not only a member of one but of various communities, typically even of communities with the same topics. E.g., both eBay and Amazon[5] are providers of marketplace communities and many people use both. For this reason there is an interest in using reputation independent from the community.

At the latest with the integration of various communities into one reputation system privacy becomes an important issue. Unfortunately reputation systems as data bases that collect information about who interacted with whom in which context will be a promising target for numerous data collectors. For this reason such information should be protected by means of technical data protection to ensure users' right of informational self-determination [9].

Privacy-enhancing user-controlled identity management [3,4] like PRIME[6] assists users platform-independent in controlling their personal data in various applications and selecting pseudonyms appropriately depending on their wish for pseudonymity and unlinkability of actions.

Reputation usually is assigned to a reputation object (e.g. a pseudonym). The interoperability of a reputation system with a user-controlled privacy-enhancing identity management needs a privacy-respecting design of reputation systems while keeping the level of trust provided by the use of reputations. In this paper we present such a reputation system that collects reputation from various communities and makes it usable for their members. Especially we try to follow the design options of a privacy-respecting reputation system for centralised Internet communities [12]. In section 2 we give an overview of the scenario and the requirements the reputation system should fulfil. Based on this analysis in section 3 we describe our system developed including implementation and evaluation.

2 Scenario

A community system offers its users the possibility to interact with each other.

The members of a community can be assisted by an identity management system that helps them to decide which pseudonym to use in which community and with which interactor. Unfortunately current identity management focuses on the typical user-service-scenario and should be extended to be also of use in communities [1] as planned for PrimeLife[7].

The members of a community can be assisted by a reputation management to select and decide with whom to interact. The reputation management covers several communities as a reputation network. Users within the reputation network can give ratings to each other based on interactions within the communities. It collects these ratings independent from the community and aggregates them to the respective member's reputation in the reputation network.

[5] http://www.amazon.com/

[6] Privacy and Identity Management for Europe (http://www.prime-project.eu/), funded by the European Union in the 6. Framework Program, 2004-2008.

[7] Privacy and Identity Management in Europe for Life (http://www.primelife.eu/), funded by the European Union in the 7th Framework Program, 2008-2011.

In our work we focus on centralised implementations of community systems currently common on the Onternet. But as common for user-controlled privacy-enhancing identity management for the reason of informational self-determination we want to give the single user the control over his reputation and assume the reputation to be stored locally at the owner's device. He has to be assisted by a reputation provider to guarantee some security requirements that will be outlined in section 2.1. The reputation for this reason should be globally the same for the user in a certain context. By the use of user-controlled privacy-enhancing identity management he can separate pseudonyms and by interoperable reputation management respective reputation for distinct contexts.

2.1 Stand-Alone Centralised Community and Reputation Systems

In a centralised community system interactions between members take place via a central community server where they are stored and globally available at. To become a member of the community a user has to register with the community server by declaring a pseudonym for use within the community. Possible examples are web forums like phpBB. If a reputation system is in place the community server also overtakes the role of a reputation server: Users have the opportunity to give ratings to interactions and based on these ratings a global reputation for every user is computed . Before an interaction members inform themselves about each other to decide whether to interact and what to expect from an interaction. According to [8] reputation systems have to provide the following protocols:

1. Centralised communication protocols that allow members to:
   - provide ratings about other interactors,
   - obtain reputation of potential interactors from the reputation server.
2. A reputation computation algorithm the reputation server uses to derive members' reputation based on received ratings, and possibly other information.

If the system follows a multilateral secure approach [10], it respects the different security requirements of all users involved. The requirements outlined in the following are the generic security requirements for reputation systems we helped to elaborate in [6] focused on our scenario.

Availability of reputation: Users of the reputation system want to access reputations as functional requirement to select interactors.

Integrity of interactions and ratings: The reputation information needs to be protected from unauthorised manipulation, in propagation and in storage.

Accountability of interactors and raters: Users want other interactors and raters to be accountable for their actions and ratings.

Completeness of ratings and reputation: Users want ratings to be given for all interactions a pseudonym participated in. The aggregated reputation should consider all ratings given.

Pseudonymity of raters and interactors: Users want to rate and interact under a pseudonym that is not necessarily linked to their real name.

Unlinkability of ratings and actions: Users want their ratings and actions to be unlinkable. Otherwise behaviour profiles of pseudonyms could be built. If the pseudonym becomes linked to a real name, as it often does in a reputation network, the profile becomes related to this real name as well.

Anonymity of users: Users want to inform themselves anonymously about others' reputation to prevent behaviour profiles of their possible interests.

Authorisability of interactions, ratings and reputation computation: Interactions should only take place between authorised members. Ratings should only be given by members if an interaction took place between them.

Confidentiality of interactions, ratings and reputations: Though a reputation system's functional requirement is to collect ratings, the aggregated reputation should only be obtainable by members. Concrete users also might want only a subset of other members to know their ratings and reputation.

There exist countless models to design possible ratings and the reputation computation algorithm [8]. As outlined in [6] the following requirements hold:

Accuracy of the reputation computation algorithm: The reputation computation algorithm has to consider all ratings given. This should also consider long-term performance. Other aspects include soliciting ratings, but also educing hidden one (e.g. lack of rating). It should be possible to distinguish between newcomers and users with bad reputation.

Trustworthiness of the raters: Existing social networks, and weighting recommendations according to the trustworthiness of the raters should be used to scale the reputation computation algorithm to the needs of the one who wants to inform himself about others' reputation.

Self-correction: If a user agrees not or no longer with certain ratings for a certain reputation object, he should correct both the trust values for the corresponding raters in the reputation computation algorithm and the reputation of the reputation object.

These requirements show the need for individual reputation in contrast to the common concept of global reputation. Due to the need of multilateral secucurity we also favoured global repuation and had to neglect the last two requirements above to reach anonymity and unlinkability of the uders involved.

2.2 Cross-Community Reputation Management

With the help of cross-community reputation a significant reputation can be built up easily in short time even in very small communities.

The reputation system should be independent from the communities the reputation it aggregates s used for. The reputation system provides the communication protocols necessary. But it might offer the communities the possibility to define their own reputation computation algorithm that should be used by the reputation system. In practice this will mean that the reputation visible in the communities will differ. For this reason the ratings given should consist of

- the concrete mark given to the reputation object,
- the context the mark is given in,
- the community the interaction took place.

The communities have to agree on appropriate contexts they are willing to exchange reputations for. The context might be the type of community and the role the reputation object has/had (e.g. a seller in a marketplace).

The following requirements that are derived by multilateral security should hold for the reputation system in addition to the scenario of one community:

Authorisability of cross-community reputation: Members within the communities have to agree on the communities their reputation is collected for.

Unlinkability of community pseudonyms: Users want to be members of different communities without being linkable.

The latter requirement seems to be a contradiction to the functional requirement of cross-community reputation and the community noted in every rating but this reveals only in which communities a user collects ratings but the pseudonyms he uses within these communities can still be unlinkable.

3 System Design

The system consists of three parts: the community server, which is realised through phpBB, a user-controlled privacy-enhancing identity management like PRIME and the reputation system, which is responsible for all the functions, which are related to the handling of the reputations.

This system design tries to fulfil both the security requirements stated for stand-alone community reputation systems as outlined in 2.1 and cross-community reputation systems as outlined in 2.2 in the sense of multilateral security.

The system uses global reputations that are stored at the users' device to give him control over both his reputation and his privacy. Our design is independent from concrete ratings and reputation computation algorithms.

We assume all communication to be secured by encryption to reach confidentiality of all ratings and actions performed. All actions and ratings have to be secured by digital signatures given under a pseudonym for integrity reasons. By the use of an identity provider accountability of the pseudonym can be given.

For the identity management a user Alice registers a basic pseudonym with an identity provider by declaration of her identity data (step 1 in Fig. 1). After verifying the data the identity provider issues a basic credential (step 2 in Fig. 1).

When Alice wants to register in a reputation network within a certain context she sends the reputation provider her basic credential (step 3 in Fig. 1). This guarantees no user is able to build up reputation under multiple pseudonyms within the same context and every user can be identified in the case of misbehaviour. The reputation provider creates a reputation pseudonym based on the basic pseudonym and sends it back to Alice (step 4 in Fig. 1).

The reputation credential contains the pseudonym and its initial reputation. The credential is a pseudonymous convertible credential [2] the user can convert

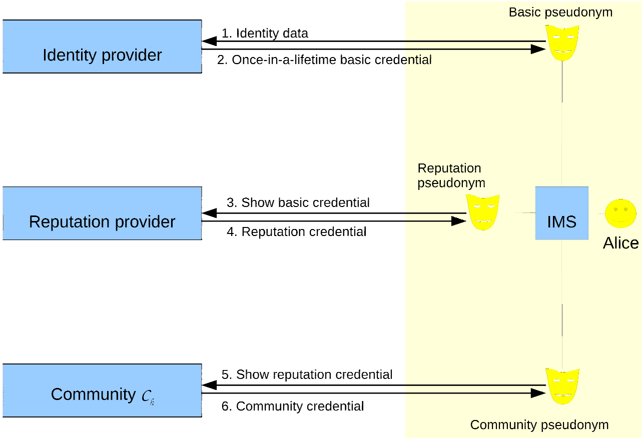


Fig. 1. Registration process enabling unlinkability of a user and his pseudonyms

to another pseudonym within the reputation network whenever he wants to reach unlinkability of actions. The credentials contain an attribute for the context, $l > 0$ attributes for the last l ratings and an attribute for the expiration date.

After the conversion of the reputation credential to a community pseudonym Alice can register this pseudonym with a community $\mathcal{C}_h$ by showing the converted credential (step 5 in Fig. 1). Thereby she agrees that she will collect reputation for her interactions in the community with the reputation network she registered with. Based on this she gets a community credential to her community pseudonym and becomes a member of the community (step 6 in Fig. 1).

By the use of these distinct pseudonyms, unlinkability of the actions performed under these pseudonyms is given initially. The only exception are Alice's reputation pseudonym and community pseudonym because Bob wants to assure that he actually gave the rating to the pseudonym he interacted with.

3.1 Reputation System

In the following we outline the design of or reputation system.

Leave Rating To leave a rating an interaction must have taken place and been finished between the respective two pseudonyms of Alice and Bob. After an interaction (step 1 in Fig. 2) Bob receives a convertible credential from the community that states that an interaction has been finished and Bob is allowed to rate Alice's pseudonym (step 2 in Fig. 2). Bob is able to convert this credential from his community pseudonym to his reputation pseudonym (step 3 in Fig. 2).

For the rating (step 3 in Fig. 2) Bob sends this credential, Alice's pseudonym and the actual rating he wants to give to Alice to the reputation provider who tests its validity and stores the rating until the update of Alice's reputation.

Update of the reputation. After a fixed number $k \geq 1$ of ratings have been given to Alice's pseudonym its reputation has to be updated by the reputation

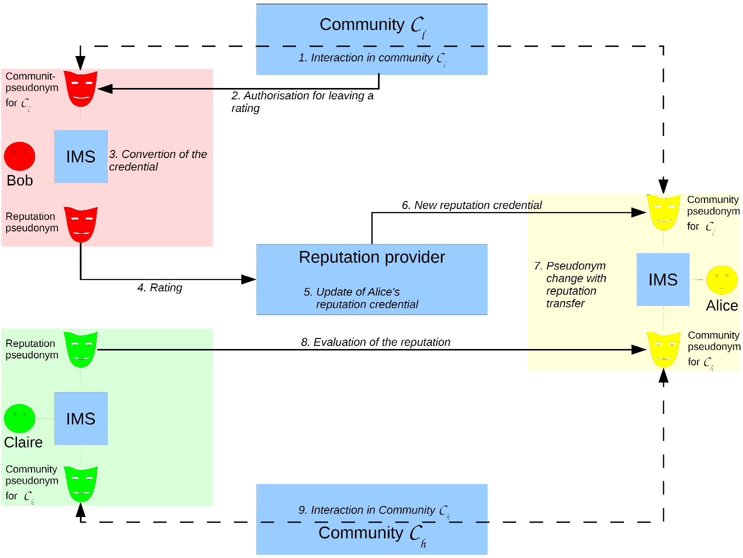

Community C_i
1. Interaction in community C_i
Communit-pseudonym for C_i
2. Authorisation for leaving a rating
IMS
3. Convertion of the credential
Bob
Reputation pseudonym
6. New reputation credential
Community pseudonym for C_i
Reputation provider
4. Rating
5. Update of Alice's reputation credential
7. Pseudonym change with reputation transfer
IMS
Alice
Reputation pseudonym
8. Evaluation of the reputation
Community pseudonym for C_h
IMS
Claire
Community pseudonym for C_h
9. Interaction in Community C_h
Community C_h

Fig. 2. System design

provider (step 5 in Fig. 2). We do not fix $k = 1$ here because according to the game-theoretical analysis in [5] it might make sense economically not to update a reputation after every rating but only after $k > 1$ ratings. This also increases Alice's unlinkability.

For the update Alice has to send her reputation credential to the reputation system. This might be either initiated by Alice or by the reputation provider. The attribute containing the reputation has to be updated in the reputation credential 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 reputation value. Only the relationship between the old and the new credential must be guaranteed by the reputation provider. Therefore in principal the calculation is possible on encrypted values if the reputation computation 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 sets of possible reputations and ratings have to be small enough to reach large enough anonymity sets. Details about this idea are outlined in [12].

For the update the reputation provider sends the new reputation credential to Alice (step 6 in Fig. 2). The old reputation credential would still be valid if it did not contain the attribute for the expiration date.

Pseudonym change with reputation transfer. To increase the unlinkability between different interactions of a user, the change of pseudonyms with reputation transfer is possible as suggested in [12] (step 7 in Fig. 2). This is realised by pseudonymous convertible credentials that allow a user to maintain his reputation but use a new pseudonym without trusting the reputation provider.

A pseudonym change only makes sense when a large number of users with the same attributes (here the same reputation if no other attributes are known) changes their pseudonym at the same time to guarantee an appropriate anonymity set. For this reason the sets of possible rating and reputation values are limited.

If Alice wants to change her pseudonym while a rating has been left at the reputation provider for her credential, it cannot 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 indirectly by issuing credentials with new expiration dates. By this he helps to collect an anonymity set of users willing to change their pseudonyms.

Evaluation of reputation. Before deciding on an interaction with a member of the community $\mathcal{C}_h$ Claire can evaluate pseudonymously its reputation after the member send her the reputation credential (step 8 in Fig. 2).

To augment the availability of the reputation a storing at the reputation server or the community server should be possible with the chance for the user to appoint authorisation to other members of the community to see the reputation.

Leaving a reputation network. Alice can always leave the community or reputation network. If she then has a reputation less than the initial reputation her identity should be revealed and banned by the identity provider to guarantee that she does not get any new basic pseudonyms she could use for a new registration in the reputation network or a community. This implements the once-in-a-lifetime-credentials introduced in [7]

3.2 The Prototype

phpBB The software phpBB was originally developed as software for forums. Therefore text-based interactions can be carried out with the help of phpBB. The framework has a centralised architecture that must be installed on a web server using PHP as script language. It supports various database schemes (MySQL, etc.). The user uses the system only with the help of a web-based interface. The basic phpBB implementation allows users to register with the community, to start and answer a thread. For a reputation system like ours where users should be rated based on interactions it is crucial that a mechanism exists, which proves that the interaction has actually happened and was finalised. Such a mechanism provides the MOD "Geocator's Feedback Ratings MOD". Besides it includes a whole reputation system in an eBay-like style we do not make use of.

Reputation system. The credentials and the required functions for handling them were implemented using the idemix-Framework[8], which is written in Java.

[8] http://www.zurich.ibm.com/security/idemix/

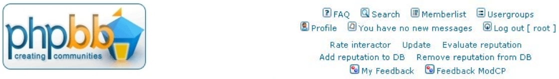


Fig. 3. Extended interface of phpBB

The reputation system is independent from the community server but can be called over links integrated in the phpBB framework. These links lead to PHP-based websites, offering different functions of the reputation system.

The websites request the users to fill in the necessary specifications like the reputation credential or the rating value. If the inputs are valid after checking by the reputation system, the PHP-scripts call a Java program implementing the respective reputation functions. The programs are either dealing on the credentials (e.g. the update function) or on one of the databases also implemented by the idemix framework (e.g. the rating function, where the rating remains in the database till the reputation object updates his reputation credential). Also the published reputation is in one of these databases. Functions to remove one's reputation and to search for other members' reputation are also existent.

The prototype does not use PRIME yet but uses the authentication methods of phpBB. Therefore the registration process takes place simultaneously in the phpBB community and the reputation system. The phpBB community could be used as usual, but the member can call the reputation functions within the phpBB interface that have been extended for this reason as illustrated in Fig. 3.

Pseudonym change. The pseudonym change is implemented in an Java-program which can be executed on the user's system without knowledge of the reputation provider, the community or other members.

3.3 Evaluation

The prototype was exemplary evaluated by a small testing group of 13 persons. One questionnaire asked for general issues of reputation systems, communities and privacy related concerns. The second questionnaire dealt with experiments made with the prototype.

The first questionnaire showed that the users pay attention that their real name is not published within communities. But users seem not to be aware of how much information of their behaviour can be collected. Only the half of the evaluators saw reputation as privacy-invasive information.

The prototype itself found general approval of the evaluators. The dealing with the system was mastered by nearly all the evaluators. Only half of the evaluators approved the concept of the pseudonym change or declared that they understood the relevance of credentials. Maybe the benefit of this function and the uses of the credentials in general have to be promoted more vigorously. The separation of reputation system and community was found to be irritating while searching for another user's reputation. While keeping the separation of the systems for the reason of unlinkability this should be invisible to the user.

4 Conclusion

In this paper the basis and preconditions to design an interoperable reputation system for Internet communities were introduced. This concept becomes more and more important with the number of communities with similar topics growing. Interoperability and the possible transfer of reputation lead to new possibilities how to deal with newcomers in communities. Although we paid special attention to the unlinkability of pseudonyms in different communities our solution still needs trust in the provider. In future research we will concentrate on an easier and more privacy-respecting handling of users' various identities and reputations.

References

1. 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)
2. Chaum, D.: Showing credentials without identification. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 241–244. Springer, Heidelberg (1986)
3. Clauß, S., Pfitzmann, A., Hansen, M., Van Herreweghen, E.: Privacy-enhancing identity management. The IPTS Report 67, 8–16 (2002)
4. Clauß, S., Köhntopp, M.: Identity management and its support of multilateral security. Computer Networks 37(2), 205–219 (2001)
5. Dellarocas, C.: Research note – how often should reputation mechanisms update a trader's reputation profile? Information Systems Research 17, 271–285 (2006)
6. ENISA. Position Paper no.2. Reputation-based Systems: A Security Analysis (October 2007), http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_pp_reputation_based_system.pdf
7. Friedman, E., Resnick, P.: The social cost of cheap pseudonyms. Journal of Economics and Management Strategy 10, 173–199 (1999)
8. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43, 618–644 (2007)
9. Mahler, T., Olsen, T.: Reputation systems and data protection law. In: eAdoption and the Knowledge Economy: Issues, Applications, Case Studies, pp. 180–187. IOS Press, Amsterdam (2004)
10. Pfitzmann, A.: Technologies for multilateral security. In: Müller, G., Rannenberg, K. (eds.) Multilateral Security for Global Communication, pp. 85–91. Addison-Wesley, Reading (1999)
11. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Communications of the ACM 43(12), 45–48 (2000)
12. 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)

Fairness Emergence through Simple Reputation*

Adam Wierzbicki and Radoslaw Nielek

Polish-Japanese Institute of Information Technology
Warsaw, Poland
adamw@pjwstk.edu.pl, radoslaw.nielek@pjwstk.edu.pl

Abstract. Trust Management is widely used to support users in making decisions in open, distributed systems. If two sellers on e-Bay have similar goods and service, similar marketing, they should also have similar income and reputation. Such an expectation can be formulated as a hypothesis: in realistic reputation (or trust management) systems, fairness should be an emergent property. The notion of fairness can be precisely defined and investigated based on the theory of equity. In this paper, we investigate the Fairness Emergence hypothesis in reputation systems and prove that in realistic circumstances, the hypothesis is valid. However, Fairness Emergence is not a universal phenomenon: in some circumstances it would be possible for one of two similar sellers to be better off. We study the sensitivity of Fairness Emergence to various aspects of a reputation systems.

1 Introduction

In distributed, open systems, where the behavior of autonomous agents is uncertain and can affect other agents' welfare, trust management is widely used. Examples of practical use of trust management are (among others) reputation systems in online auctions and Peer-to-Peer file sharing systems.

From the point of view of the agents who participate in transactions and use a reputation system to cope with uncertainty or risk, the fairness of such a system is particularly important. Consider the example of online auctions. While the owner of the auction system might care only for an increase of the transaction volume, the buyers or sellers expect that if they behave as fairly as their competitors, they should have a similarly high reputation. In other words, the users of a reputation system expect that the reputation system should treat them as fairly as possible.

This intuitive reasoning leads to the formulation of a hypothesis: if a reputation system works better, then the utilities of similar users should become more equal. This hypothesis could also be formulated differently: in successful reputation (or trust management) systems, fairness should be an emergent property. We shall refer to this hypothesis as the Fairness Emergence (FE) hypothesis. In this paper, we verify the FE hypothesis.

* This research has been supported by the Polish Ministry of Science grant N N516 4307 33.

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 79–89, 2008.

The FE hypothesis is related to another question: should fairness be a goal of trust management systems? If so, how can this goal be achieved in practice? In order to consider fairness, it becomes necessary to define it precisely. In this work, fairness is defined based on a strong theoretical foundation: the theory of equity. The concept of fairness in trust management systems and the theory of equity are discussed in the next section. Section 3 describes a simulation approach for verifying the FE hypothesis, based on the fairness criteria introduced in section 2. Section 4 describes the simulation results and the sensitivity of fairness emergence to various aspects of a reputation system. Section 5 concludes the paper.

2 Considering Fairness in Trust Management Systems

Reputation systems have usually been studied and evaluated using the utilitarian paradigm that originates from research on the Prisoner's Dilemma. Following the work of Axelrod [1], a large body of research has considered the emergence of cooperation. The introduction of reputation has been demonstrated as helpful to the emergence of cooperation[1]. In the Prisoner's Dilemma, the sum of payoffs of two agents is highest when both agents cooperate. This fact makes it possible to use the sum of payoffs as a measure of cooperation in the iterated Prisoner's Dilemma. This method is an utilitarian approach to the evaluation of reputation systems [2][3][4]. In most research, a reputation system is therefore considered successful when the sum of utilities of all agents in the distributed system is highest. Note that the utilitarian paradigm is used even if the simulation uses a more complex model of agent interaction than the Prisoner's Dilemma.

The use of Prisoner's Dilemma allows for an implicit consideration of agent fairness, while the sum of utilities is considered explicitly. Yet, in a more realistic setting, the assumptions of the Prisoner's Dilemma may not be satisfied, and it is possible to point out cases when the utilitarian approach fails to ensure fairness: in an online auction system, a minority of agents can be constantly cheated, while the sum of utilities remains high. A notable example of explicit consideration for fairness of reputation systems is the work of Dellarocas [3]. An attempt to demonstrate that explicit consideration of fairness leads to different results in the design and evaluation of reputation systems has been made in [5].

2.1 Theory of Equity

In this work, the concept of system fairness is identified with *distributive fairness*, a sense narrower than social justice [6]. The understanding of the concept of fairness in this paper is based on the theory of equity. The Lorenz curve is obtained by taking the outcomes of all agents that participate in a distribution and ordering them from worst to best. Let us denote this operation by a vector function $(y) = [\theta_1(y), ..., \theta_n(y)]$ of the outcome vector y (the outcomes can be utilities of

[1] However, note that the existence of reputation information is a modification of the original Prisoner's Dilemma. Axelrod has explicitly ruled out the existence of reputation information in his definition of the game.

agents in a distributed system). Then, the cumulative sums of agents' utilities are calculated: starting from the utility of the worst agent (θ_1), then the sum of utilities of the worst and the second worst (θ_2), and so on, until the sum of all agents' utilities. Let us denote it as θ_n. The equal distribution line is simply a straight line connecting the points (1, θ_1) and (n, θ_n). The area between the two curves, denoted by S, can be seen as a measure of inequality of the agent's utilities. The objective of distributive fairness is to minimize this inequality, making the Lorenz curve as close to the equal distribution line as possible.

The area between the Lorenz curve and the equal distribution line can be simply calculated and used as a computable measure of inequality. It can be shown that minimizing this measure leads to fair distributions [7]. The Gini coefficient (frequently used in economics) is the area S normalized by θ_n: $Gini = \frac{S}{2\theta_n}$. Note that minimizing the Gini coefficient to obtain fair distributions can lead to worse total outcomes (sums of all agent's utilities) - this drawback can be overcome by using a different fairness measure: the area under the Lorenz curve (also equal to $\frac{n\theta_n}{2} - S$).

Using the theory of equity, the Fairness Emergence hypothesis can be reformulated as follows: *in successful trust management systems, the distribution of similar agents' utilities should become more equitable.* The fairness criteria described in this section can be used to check whether the distribution is more equitable.

2.2 Laboratory Evaluation of Trust Management Systems

In a real-world setting, users of trust management systems would be expected to have quite varied levels of utility (perhaps even incomparable ones). How, then, do we expect a trust management system to realize a goal of fairness? And how can the Fairness Emergence hypothesis be true?

This concern is based on a frequent misconception that mistakes equality for fairness. If a trader in an Internet auction house has better goods, provides better services and has better marketing than other traders, it is perfectly fair that he should have a larger transaction volume and a larger revenue. On the other hand, if we have two honest traders that have comparable goods, services, and marketing, yet they have very unequal reputation and transaction volumes, surely something is wrong in the way the trust management system works.

Therefore, when all other factors can be excluded (equivalent to the ceteris paribus assumption from economics), fairness can be identified with distributional fairness. In a laboratory setting, such conditions can be satisfied and we can design trust management systems that realize the goal of fairness, even in the presence of adversaries.

3 Verifying the Fairness Emergence Hypothesis by Simulation

To verify the Fairness Emergence hypothesis, we have used a simulation experiment. The FE hypothesis would hold if we could establish that the reputation system causes an increase of the equity of the distribution of utilities. In particular, we will be interested to study the impact of the quality of the reputation system on the equity of utility distributions.

The simulator is based on the Repast 3.1 platform [12] and resembles an Internet auction system. In the design of the simulator, we had to make a decision about a sufficiently realistic, yet not too complex model of the auction system, of user behavior, and of the reputation system. We chose to simulate the reputation system and the behavior of user almost totally faithfully (the only simplification is that we use only positive and negative feedbacks).

The auction system, on the other hand, has been simplified. We simulate the selection of users using random choice of a set of potential sellers. The choosing user (the buyer) selects one of the sellers that has the highest reputation in the set (and then checks if the selected one has a reputation that is higher than a threshold).

3.1 Agent Behavior

In our simulator, a number of agents interact with each other. There are two types of agents in the system: honest and dishonest agents. Dishonest agents model adversaries. To test the FE hypothesis, we shall be interested in the fairness of utility distributions of honest agents. The payoffs of honest and dishonest agents will also be compared.

When an agent carries out a transaction, it must make three decisions. The first decision concerns the choice of a transaction partner (seller) and whether or not to engage in the transaction. The agent chooses his partner from a randomly selected set of k other agents (in the simulations presented here, k has been equal to 3 or 1). From this set, the agent with the highest reputation is chosen. However, if the highest reputation is lower than a threshold p_{min}^{choice} (honest agents choose partners with reputation at least 0.45, and dishonest agents: 0.3) , then the choosing agent will not engage in any transaction. If the best agent's reputation is sufficiently high, the choosing agent will engage in the transaction with a certain probability p (in the simulations presented here, this probability was 1).

The second decision concerns the agent's behavior in the transaction. This decision can be based on a game strategy that can take into consideration the agent's own reputation as well as the reputation of his partner, the transaction history and other information. We decided to use the famous Tit-for-tat strategy developed by Rapaport but extended with using a reputation threshold: if two agents meet for the first time and the second agents' reputation is below p_{min}^{game} , the first agent defects. The strategy used in the simulations presented here has also been based on the threshold p_{min}^{cheat}. In the case when the partner's reputation is higher than p_{min}^{cheat}, the agent would act fairly; otherwise, it would cheat with a certain probability c. In the simulations presented here, honest agents had a cheating probability of 0, while dishonest agents had a cheating probability of 0.2 and a reputation threshold of 0 - meaning that dishonest agents cheated randomly with a probability of 0.2.

The third decision of the agent concerns the sending of reports. For positive and negative reports, an agent has separate probabilities of sending the report. In the simulations presented here, the probability of sending a positive report, p^{+}_{rep} was 1.0, while the probability of sending a negative report p^{-}_{rep} varied from 0 to 1. This choice is based on the fact that in commonly used reputation systems [4], the frequency of positive reports is usually much higher than of negative reports. In the simulation it is also possible to specify a number of agents that never send reports. This behavior is independent of the honesty or dishonesty of agents.

3.2 Simulation Experiments

In all simulations, there was a total of 1500 agents, out of which 1050 where honest and 450 were dishonest. While the proportion of dishonest agents is high, they cheat randomly and at a low probability - so a dishonest agent is really a "not totally honest agent". Also, considering that frauds in Internet auctions are among the most frequent digital crimes today, and considering that cheating in an auction may be more frequent than fraud - it may be sending goods that are of worse quality than advertised - this proportion of dishonest agents seems realistic.

The simulator can compute reputations using all available feedbacks. The results of the simulation include: the reputations of individual agents and the total utilities (payoffs from all transactions) of every agent. In the simulations presented here, an agent's reputation is computed as the proportion of the number of positive reports about the agent to the number of all reports.

All simulations were made using pseudo-random numbers, therefore the Monte Carlo method is used to validate statistical significance. For each setting of the simulation parameters, 50 repeated runs were made, and the presented results are the averages and 95% confidence intervals for every calculated criterion. The confidence intervals were calculated using the t-Student distribution.

We decided to use transaction attempts instead of the number of successful transaction as a stop condition because we believe that an agent would consider each transaction attempt as an expense, and the reputation system would have to work well after as few transaction attempts as possible. In most presented simulations for each turn, 500 transaction attempts have been made.

For each simulation, the first 20 turns have been used to warm-up the reputation system. It means that the payoffs are not recorded but an agents' reputation is modified by positive and negative reports. This method has been used to model the behavior of a real reputation system, where the system has available a long history of transactions. Simulating the reputation system without a warm-up stage would therefore be unrealistic.

4 Simulation Results

To verify the Fairness Emergence hypothesis, we have been interested to investigate the impact of a reputation system on the equity of the agent utility distribution. Equity of utility distributions has been measured using fairness criteria based on the theory of equity; however, other criteria such as the sum of agent utilities are considered as well. The simulations revealed that the Fairness

Emergence hypothesis holds in several cases, but not universally; therefore, we have investigated the sensitivity of fairness emergence to various factors that influence the quality of the reputation system.

4.1 Fairness Emergence in the Long Term

The first studied effect has been the emergence of fairness in the long term. In the simulation experiment, we have measured the Gini coefficient and have run the simulation until the Gini coefficient stabilized. This experiment has been repeated using three scenarios: in the first one, the agents did not use any reputation system, but selected partners for transactions randomly. In the second experiment, the reputation system was used, but agents submitted negative reports with the probability of 0.2. In the third experiment, negative reports have always been submitted.

The results of the three experiments are shown on Figure 1. The Figure plots the average Gini coefficient of honest agents from 50 simulation runs against the number of turns of the simulation. It can be seen that when agents do not use the reputation system, the Gini coefficient stabilizes for a value that is almost twice larger than the value of Gini that is obtained when reputation is used. Furthermore, there is a clear effect of increasing the frequency of negative feedbacks: the Gini coefficient decreases faster and stabilizes at a lower value when $p_{rep}^{-} = 1$. The initial growth of the Gini coefficient from 0 is due to the fact that at the beginning of the simulation, the distribution of honest agent utilities is equal (during the warm-up stage, utilities of agents are not recorded. All agents start with a zero utility after warm-up completes.)

The result of this experiment seems to be a confirmation of the FE hypothesis. The distributions of honest agents' utilities have a lower Gini coefficient (and a higher total sum of utilities) when the reputation system is used. Yet, the problem here is that in realistic auction systems, most agents only have a small

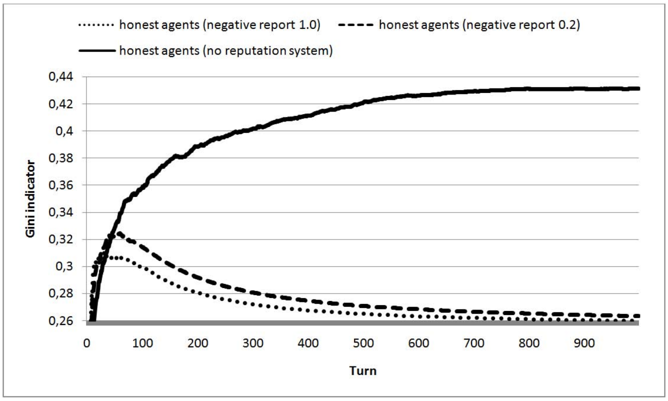


Fig. 1. Fairness Emergence in the long term

number of successful transactions, because they use the system infrequently. In our simulation, new agents did not join the system (although the number of agents was large). The average number of successful transactions of an agent has been about 270, which is much lower than the number of agents; this means that as in a real auction system, the chance of repeated encounters was low. However, this number is still large. The simulations were continued until a stable state was reached; in practical reputation systems, such a situation would not be likely to occur because of the influx of new agents and the inactivity of old ones. For that reason, we have decided to investigate the FE hypothesis in the short term, or in unstable system states.

4.2 Fairness Emergence in the Short Term

The simulation experiments studied in the rest of this paper have been about 8 times shorter than the long-term experiments. For these experiments, the number of successful transactions of an average agent was about 60. Figure 2 shows the Gini coefficient of the distributions of honest agents' utilities. On the x axis, the frequency of sending negative reports by honest agents is shown (dishonest agents always sent negative reports). The results show that for low negative report frequencies fairness emerges more slowly. Increasing the quality of a reputation system reduces the time needed for fairness emergence. This effect is apparent very quickly, even after 50 turns of simulation. From now on, fairness emergence in the short term is studied more closely to verify whether the improvement of reputation system quality will cause fairness emergence. In other words, until now we considered fairness emergence with time, and now we shall consider the effect of the reputation system's quality on fairness emergence. All further experiments have been made in the short term, outside of the stable state of the system.

4.3 Effect of Better Usage of Reputation

The usage of reputation by agents had a particularly strong influence on the emergence of fairness. In our simulations, agents chose a seller with the highest

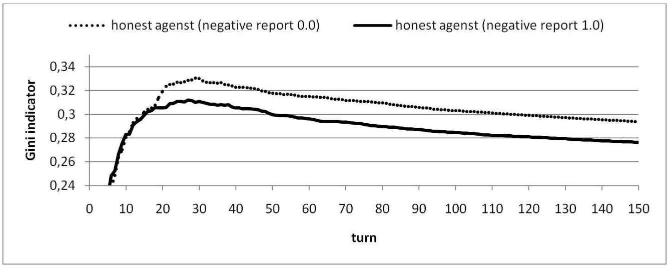


Fig. 2. Fairness Emergence in the short term

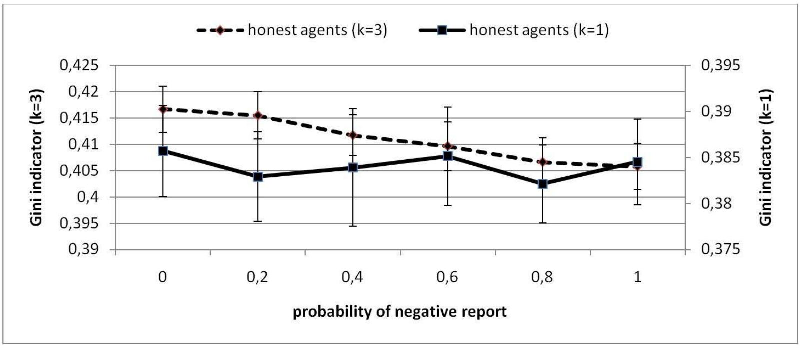


Fig. 3. Effect of increased choice on Gini coefficient

reputation. If k=1, then the transaction partner was chosen at random and only the threshold $p_(min)^(game)$ was used to consider reputation. If k=3, it was less likely that an agent with lower reputation would be chosen as a transaction partner. These two scenarios correspond to the real life situation of buyers who are able to select sellers from a larger set, based on their reputation; on the other hand, it could be possible that the choice is low, because only one seller has the required goods or services.

We have considered the two scenarios while investigating the impact of the frequency of feedbacks on the reputation system. It turns out that increasing the choice of agents is necessary for the emergence of fairness. Figure 3 shows the effect of increasing the frequency of negative feedback on the Gini coefficient of honest agents. The figure shows two lines that correspond to the scenarios of k=1 and k=3. It can be seen that if the choice of agents on the basis of reputation is possible, then the increase in the number of feedbacks leads to a decrease of the Gini coefficient.

Figure 4 shows the effect of increased choice and varying negative feedback frequency on the sum of honest agents' utilities. It can be seen that once again, enabling the choice of partners based on reputation has a positive effect on the

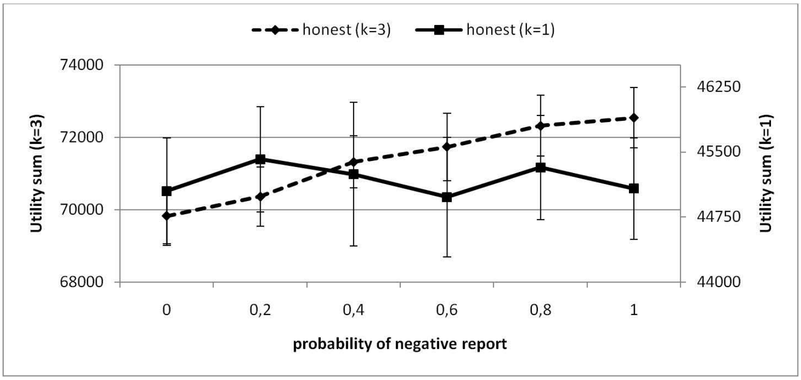


Fig. 4. Effect of increased choice on sum of utilities

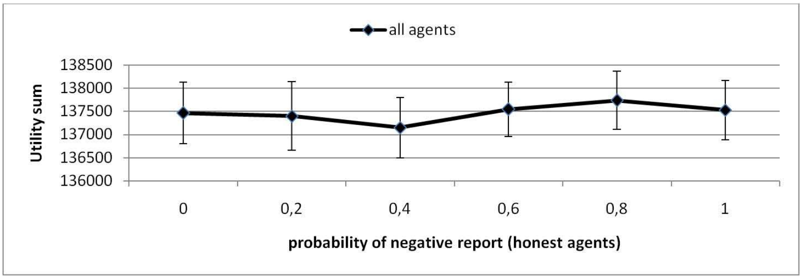


Fig. 5. Effect of increased feedback on sum of utilities of all agents

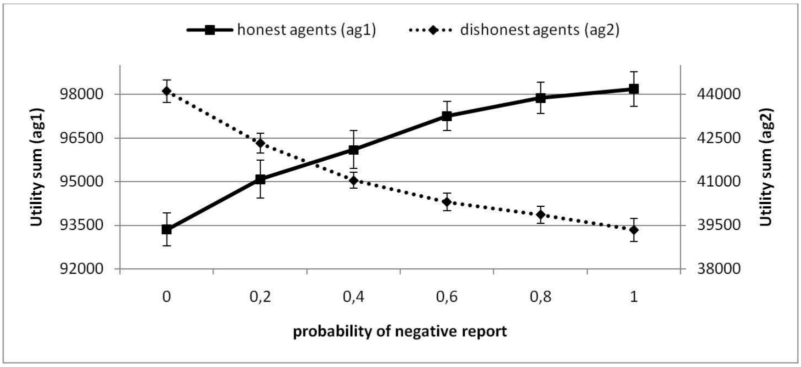


Fig. 6. Effect of increased feedback on honest and dishonest agents' utilities

welfare of honest agents. For k=3, honest agents overall had a higher sum of utilities than for k=1, and this sum increased when the frequency of negative reports increased. This also explains why the Gini coefficient of honest agents for k=1 was lower than for k=3. Since the sum of utilities was lower for k=1, the Gini coefficient could also be lower, although this does not mean that the distribution of utilities for k=1 was more equitable than for k=3.

4.4 Effect of Better Feedback

Better feedback is a prerequisite for increasing the quality of a reputation system. For that reason, we have chosen to investigate the effect of increased feedback on the emergence of fairness. As has been explained previously, the frequency of negative feedback has been varied from 0 to 1.

Figure 5 shows the effect of increasing negative feedback on the sum of utilities of all agents. It turns out that the total sum was not affected by the increase. This seems to be a paradox, since we are using the iterated Prisoner's Dilemma as a model of our auction system. In the Prisoner's Dilemma, increased fairness of agents results in an increased sum of all utilities. And increasing negative feedbacks from 0 to 1 should result in decreasing the ability of dishonest agents to cheat.

This experiment also shows that even assuming the use of a Prisoner's Dilemma as a model of a transaction, the use of the sum of all agents' utilities (the utilitarian

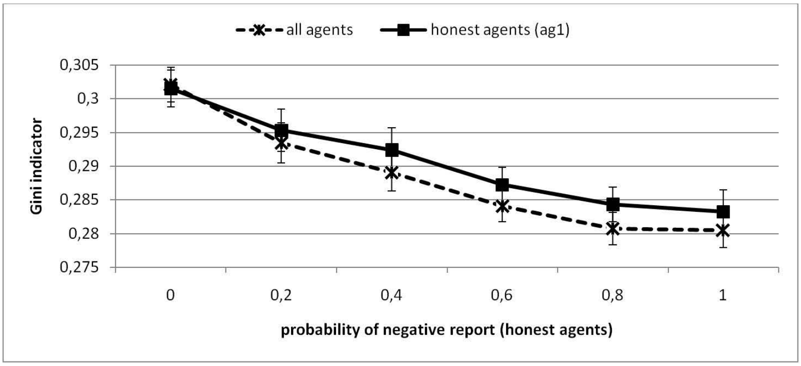


Fig. 7. Effect of increased feedback on Gini coefficient

paradigm) would lead to a wrong conclusion that the system behavior is not affected. From the utilitarian point of view, the reputation system works equally well when the frequency of negative reports is 0, as when it is equal to 1.

Figure 6 shows that this is not the case. The sum of utilities of honest agents increases, as negative feedbacks are sent more frequently. On the other hand, the sum of utilities of dishonest agents drops.

Figure 7 shows effect of increased negative feedback frequency on the Gini coefficient. Note that the effect is statistically significant for the variation of from 0 to 1 (also from 0.4 to 1). Note that these simulations have been made in the short term and that together with the results about the sum of utilities, they prove the FE hypothesis: increasing the quality of the reputation system does indeed lead to more equitable distribution of honest agents' utilities, as the hypothesis suggested.

5 Conclusion

The Fairness Emergence hypothesis may be viewed as a theoretical concept that is similar to the well-known "evolution of cooperation". On the other hand, it has an inherent practical value. First, if the FE hypothesis would not be true, then a reputation (or trust management) system would allow the existence of a degree of unfairness between similar agents. Such a situation would be highly undesirable from the point of view of users of trust management systems, leading to a disincentive of their usage. Second, if the FE hypothesis holds, then the problem of ensuring fairness in an open, distributed system without centralized control may have found a practical solution: it would suffice to use a good trust management system in order to provide fairness.

We have shown that the Fairness Emergence hypothesis applies in realistic conditions: in the presence of adversaries and in an unstable state of the system. Yet, this work also shows that the FE hypothesis does not apply universally. In particular, fairness emergence does not occur (or is very weak) if very few negative feedbacks are received by the reputation system, and also if the users of a reputation system do not have enough choice of transaction partners with a good enough reputation (this implies that if dishonest agents would be a large fraction of the population, fairness could not emerge).

From these results we can draw the following conclusions:

1. Trust Management systems should explicitly consider fairness in their evaluation
2. Special Trust Management systems need to be designed in order to guarantee fairness emergence.

In particular, increasing the frequency of negative feedbacks and the choice of transaction partners had the highest impact on fairness in our research.

Further research is necessary to establish the sensitivity of the FE hypothesis to various attacks on reputation systems, particularly to discrimination and whitewashing attacks. It is also necessary to investigate Fairness Emergence under other reputation algorithms. Furthermore, it would be desirable to investigate the emergence of fairness in more general trust management systems, for example in systems that make use of risk in decision support.

References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks, Ph.D. Dissertation, Massachusetts Institute of Technology (2003)
3. Dellarocas, C.: Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In: Proc. of the 2nd ACM Conference on Electronic Commerce, Minneapolis, MN, USA, October 17-20 (2000)
4. Morzy, M., Wierzbicki, A.: The Sound of Silence: Mining Implicit Feedbacks to Compute Reputation. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 365–376. Springer, Heidelberg (2006)
5. Wierzbicki, A.: The Case for Fairness of Trust Management. In: Proc. 3rd International Workshop on Security and Trust Management (STM 2007). ENTCS. Elsevier, Amsterdam (to appear, 2008)
6. Rawls, J.: The Theory of Justice. Harvard Univ. Press (1971)
7. Kostreva, M.M., Ogryczak, W., Wierzbicki, A.: Equitable aggregations and multiple criteria analysis. European Journal of Operational Research 158, 362–377 (2004)
8. Tan, Y., Thoen, W., Gordijn, J.: Modeling controls for dynamic value exchanges in virtual organizations. In: Jensen, C., Poslad, S., Dimitrakos, T. (eds.) iTrust 2004. LNCS, vol. 2995, pp. 236–250. Springer, Heidelberg (2004)
9. Gordijn, J., Akkermans, H.: Designing and evaluating e-Business models. IEEE Intelligent Systems 16, 11–17 (2001)
10. Wierzbicki, A.: Trust Enforcement in Peer-to-Peer Massive Multi-player Online Games. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4276, pp. 1163–1180. Springer, Heidelberg (2006)
11. Elgesem, D.: Normative Structures in Trust Management. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 48–61. Springer, Heidelberg (2006)
12. Repast Organization for Architecture and Development (2003), `http://repast.sourceforge.net`

Combining Trust and Reputation Management for Web-Based Services

Audun Jøsang[1], Touhid Bhuiyan[2], Yue Xu[2], and Clive Cox[3]

[1] UniK Graduate School, University of Oslo, Norway
josang@unik.no
[2] Faculty of Information Technology, QUT, Brisbane, Australia
t.bhuiyan@qut.edu.au, yue.xu@qut.edu.au
[3] Rummble.com, Cambridge, England
clive.cox@rummble.com

Abstract. Services offered and provided through the Web have varying quality, and it is often difficult to assess the quality of a services before accessing and using it. Trust and reputation systems can be used in order to assist users in predicting and selecting the best quality services. This paper describes how Bayesian reputation systems can be combined with trust modeling based on subjective logic to provide an integrated method for assessing the quality of online services. This will not only assist the user's decision making, but will also provide an incentive for service providers to maintain high quality, and can be used as a sanctioning mechanism to discourage deceptive and low quality services.

1 Introduction

Online trust and reputation systems are emerging as important decision support tools for selecting online services and for assessing the risk of accessing them. We have previously proposed and studied Bayesian reputation systems [6,7,8,12] and trust models based on subjective logic [4,5,10]. Binomial Bayesian reputation systems normally take ratings expressed in a discrete binary form as either positive (e.g. *good*) or negative (e.g. *bad*). Multinomial Bayesian reputation systems allow the possibility of providing ratings with discrete graded levels such as e.g. *mediocre - bad - average - good - excellent* [8]. It is also possible to use continuous ratings in both binomial and multinomial reputation systems [9]. Multinomial models have the advantage that scores can distinguish between the case of polarised ratings (e.g. a combination of strictly good and bad ratings) and the case of only average ratings.

Trust models based on subjective logic are directly compatible with Bayesian reputation systems because a bijective mapping exists between their respective trust and reputation representations. This provides a powerful basis for combining trust and reputation systems for assessing the quality of online services.

A general characteristic of reputation systems is that they provide global reputation scores, meaning that all the members in a community will see the same reputation score for a particular agent. On the other hand, trust systems can in general be used to derive local and subjective measures of trust, meaning that different agents can derive different trust in the same entity. Another characteristic of trust systems is that they

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 90–99, 2008.

Table 1. Possible combinations of local/global scores and transitivity/no transitivity

	Private Scores	**Public Scores**
Transitivity	Trust systems, e.g. Rummble.com	Public trust systems, e.g. PageRank
No Transitivity	Private reputation systems, e.g. customer feedback analysis	Reputation systems, e.g. eBay.com

can analyse multiple hops of trust transitivity Reputation systems on the other hand normally compute scores based on direct input from members in the community which is not based on transitivity. Still there are systems that have characteristics of being both a reputation system and a trust system. The matrix below shows examples of the possible combinations of local and global scores, and trust transitivity or not.

In this paper we describe a framework for combining these forms of trust and reputation systems. Because Bayesian reputation systems are directly compatible with trust systems based on subjective logic, they can be seamlessly integrated. This provides a powerful and flexible basis for online trust and reputation management.

2 The Dirichlet Reputation System

Reputation systems collect ratings about users or service providers from members in a community. The reputation centre is then able to compute and publish reputation scores about those users and services. Fig.1 illustrates a reputation centre where the dotted arrow indicate ratings and the solid arrows indicate reputation scores about the users.

Multinomial Bayesian systems are based on computing reputation scores by statistical updating of Dirichlet Probability Density Functions (PDF), which therefore are called Dirichlet reputation systems [8,9]. The *a posteriori* (i.e. the updated) reputation score is computed by combining the *a priori* (i.e. previous) reputation score with new ratings.

In Dirichlet reputation systems agents are allowed to rate others agents or services with any level from a set of predefined rating levels, and the reputation scores are not static but will gradually change with time as a function of the received ratings. Initially, each agent's reputation is defined by the base rate reputation which is the same for all agents. After ratings about a particular agent have been received, that agent's reputation will change accordingly.

Let there be k different discrete rating levels. This translates into having a state space of cardinality k for the Dirichlet distribution. Let the rating level be indexed by i. The aggregate ratings for a particular agent are stored as a cumulative vector, expressed as:

$$\vec{R} = (\vec{R}(L_i) \mid i = 1 \ldots k) \ . \tag{1}$$

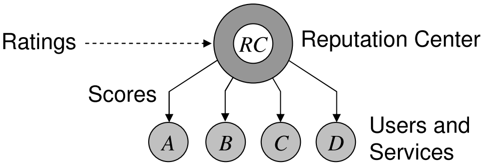


Fig. 1. Simple reputation system

This vector can be computed recursively and can take factors such as longevity and community base rate into account [8]. The most direct form of representing a reputation score is simply the aggregate rating vector $\vec{R}_y$ which represents all relevant previous ratings. The aggregate rating of a particular level i for agent y is denoted by $\vec{R}_y(L_i)$.

For visualisation of reputation scores, the most natural is to define the reputation score as a function of the probability expectation values of each rating level. Before any ratings about a particular agent y have been received, its reputation is defined by the common base rate $\vec{a}$. As ratings about a particular agent are collected, the aggregate ratings can be computed recursively [8,9] and the derived reputation scores will change accordingly. Let $\vec{R}$ represent a target agent's aggregate ratings. Then the vector $\vec{S}$ defined by:

$$\vec{S}_y : \left(\vec{S}_y(L_i) = \frac{\vec{R}_y(L_i) + C\vec{a}(L_i)}{C + \sum_{j=1}^{k} \vec{R}_y(L_j)}; \mid i = 1 \ldots k \right) . \tag{2}$$

is the corresponding multinomial probability reputation score. The parameter C represents the non-informative prior weight where $C = 2$ is the value of choice, but larger value for the constant C can be chosen if a reduced influence of new evidence over the base rate is required.

The reputation score $\vec{S}$ can be interpreted like a multinomial probability measure as an indication of how a particular agent is expected to behave in future transactions. It can easily be verified that

$$\sum_{i=1}^{k} \vec{S}(L_i) = 1 . \tag{3}$$

While informative, the multinomial probability representation can require considerable space on a computer screen because multiple values must be visualised. A more compact form can be to express the reputation score as a single value in some predefined interval. This can be done by assigning a point value ν to each rating level L_i, and computing the normalised weighted point estimate score σ.

Assume e.g. k different rating levels with point values $\nu(L_i)$ evenly distributed in the range [0,1] according to $\nu(L_i) = \frac{i-1}{k-1}$. The point estimate reputation score of a reputation $\vec{R}$ is then:

$$\sigma = \sum_{i=1}^{k} \nu(L_i)\vec{S}(L_i) . \tag{4}$$

A point estimate in the range [0,1] can be mapped to any range, such as 1-5 stars, a percentage or a probability.

Bootstrapping a reputation system to a stable and conservative state is important. In the framework described above, the base rate distribution $\vec{a}$ will define initial default reputation for all agents. The base rate can for example be evenly distributed over all rating levels, or biased towards either negative or a positive rating levels. This must be defined when setting up the reputation system in a specific market or community.

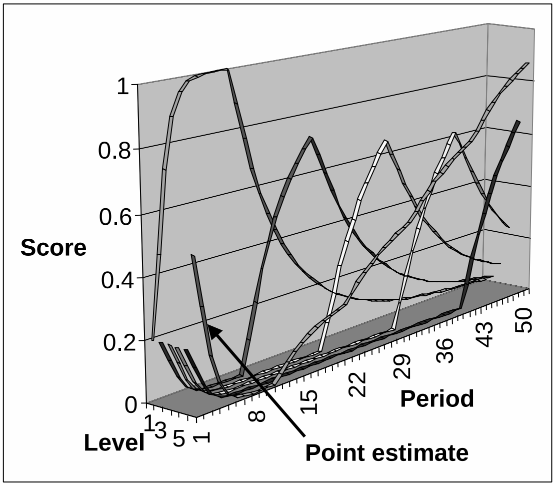


Fig. 2. Scores and point estimate during a sequence of varying ratings

As an example we consider five discrete rating levels, and the following sequence of ratings:

Periods 1 - 10:	L1 Mediocre
Periods 11 - 20:	L2 Bad
Periods 21 - 30:	L3 Average
Periods 31 - 40:	L4 Good
Periods 41 - 50:	L5 Excellent

The longevity factor is $\lambda = 0.9$, and the base rate is dynamic [8,9]. The evolution of the scores of each level as well as the point estimate are illustrated in Fig.2.

In Fig.2 the multinomial reputation scores change abruptly between each sequence of 10 periods. The point estimate first drops as the score for L1 increases during the first 10 periods. After that the point estimate increases relatively smoothly during the subsequent 40 periods. Assuming a dynamic base rate and an indefinite series of L5 (Excellent) ratings, the point estimate will eventually converge to 1.

3 Trust Models Based on Subjective Logic

Subjective logic[1,2,3] is a type of probabilistic logic that explicitly takes uncertainty and belief ownership into account. Arguments in subjective logic are subjective opinions about states in a state space. A binomial opinion applies to a single proposition, and can be represented as a Beta distribution. A multinomial opinion applies to a collection of propositions, and can be represented as a Dirichlet distribution.

Subjective logic defines a trust metric called *opinion* denoted by $\omega_X^A = (\vec{b}, u, \vec{a})$, which expresses the relying party A's belief over a state space X. Here $\vec{b}$ represents belief masses over the states of X, and u represent uncertainty mass where $\vec{b}, u \in [0,1]$ and $\sum \vec{b} + u = 1$. The vector $\vec{a} \in [0,1]$ represents the base rates over X, and is used for computing the probability expectation value of a state x as $E(x) = \vec{b}(x) + \vec{a}(x)u$, meaning that $\vec{a}$ determines how uncertainty contributes to $E(x)$. Binomial opinions are expressed as $\omega_x^A = (b, d, u, a)$ where d denotes disbelief in x. When the statement x for example says *"David is honest and reliable"*, then the opinion can be interpreted as reliability trust in David. As an example, let us assume that Alice needs to get her car serviced, and that she asks Bob to recommend a good car mechanic. When Bob recommends David, Alice would like to get a second opinion, so she asks Claire for her opinion about David. This situation is illustrated in fig. 3 below where the indexes on arrows indicates the order in which they are formed.

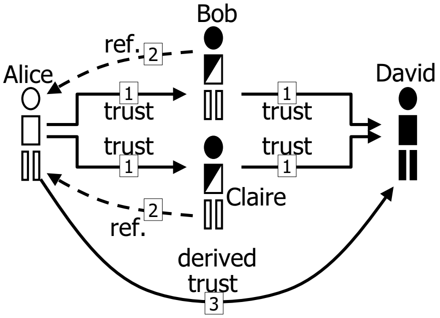


Fig. 3. Deriving trust from parallel transitive chains

When trust and referrals are expressed as subjective opinions, each transitive trust path Alice→Bob→David, and Alice→Claire→David can be computed with the *transitivity operator*[1], where the idea is that the referrals from Bob and Claire are discounted as a function Alice's trust in Bob and Claire respectively. Finally the two paths can be combined using the cumulative or averaging fusion operator. These operators form part of *Subjective Logic* [2,3], and semantic constraints must be satisfied in order for the transitive trust derivation to be meaningful [10]. Opinions can be uniquely mapped to beta PDFs, and in this sense the fusion operator is equivalent to Bayesian updating. This model is thus both belief-based and Bayesian.

A trust relationship between A and B is denoted as [A:B]. The transitivity of two arcs is denoted as ":" and the fusion of two parallel paths is denoted as "$\diamond$". The trust network of Fig.3 can then be expressed as:

$$[A, D] = ([A, B] : [B, D]) \diamond ([A, C] : [C, D]) \tag{5}$$

The corresponding transitivity operator for opinions denoted as "$\otimes$" and the corresponding fusion operator as "$\oplus$". The mathematical expression for combining the opinions about the trust relationships of Fig.3 is then:

$$\omega_D^A = (\omega_B^A \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C) \tag{6}$$

Arbitrarily complex trust networks can be analysed with TNA-SL which consists of a network exploration method combined with trust analysis based on subjective logic

[1] Also called the discounting operator.

[4,5]. The method is based on simplifying complex trust networks into a directed series-parallel graph (DSPG) before applying subjective logic calculus.

4 Combining Trust and Reputation

A bijective mapping can be defined between multinomial reputation scores and opinions, which makes it possible to interpret these two mathematical representations as equivalent. The mapping can symbolically be expressed as:

$$\omega \longleftrightarrow \vec{R} \tag{7}$$

This equivalence which is presented with proof in [3] is expressed as:

Theorem 1. Equivalence Between Opinions and Reputations
Let $\omega = (\vec{b}, u, \vec{a})$ be an opinion, and $\vec{R}$ be a reputation, both over the same state space X so that the base rate $\vec{a}$ also applies to the reputation. Then the following equivalence holds [3]:

For $u \neq 0$:

$$\begin{cases} \vec{b}(x_i) = \frac{\vec{R}(x_i)}{C + \sum_{i=1}^{k} \vec{R}(x_i)} \\ \\ u \quad = \frac{C}{C + \sum_{i=1}^{k} \vec{R}(x_i)} \end{cases} \Leftrightarrow \begin{cases} \vec{R}(x_i) \qquad = \frac{C\vec{b}(x_i)}{u} \\ \\ u + \sum\limits_{i=1}^{k} \vec{b}(x_i) = 1 \end{cases} \tag{8}$$

For $u = 0$:

$$\begin{cases} \vec{b}(x_i) = \eta(x_i) \\ \\ u \quad = 0 \end{cases} \Leftrightarrow \begin{cases} \vec{R}(x_i) \quad = \eta(x_i) \sum\limits_{i=1}^{k} \vec{R}(x_i) = \eta(x_i)\infty \\ \\ \sum\limits_{i=1}^{k} m(x_i) = 1 \end{cases} \tag{9}$$

The case $u = 0$ reflects an infinite amount of aggregate ratings, in which case the parameter η determines the relative proportion of infinite ratings among the rating levels. In case $u = 0$ and $\eta(x_i) = 1$ for a particular rating level x_i, then $\vec{R}(x_i) = \infty$ and all the other rating parameters are finite. In case $\eta(x_i) = 1/k$ for all $i = 1 \ldots k$, then all the rating parameters are equally infinite. As already indicated, the non-informative prior weight is normally set to $C = 2$.

Multinomial aggregate ratings can be used to derive binomial trust in the form of an opinion. This is done by first converting the multinomial ratings to binomial ratings according to Eq.(10) below, and then to apply Theorem 1.

Let the multinomial reputation model have k rating levels $x_i; i = 1, \ldots k$, where $\vec{R}(x_i)$ represents the ratings on each level x_i, and let σ represent the point estimate reputation score from Eq.(4). Let the binomial reputation model have positive and negative ratings r and s respectively. The derived converted binomial rating parameters (r, s) are given by:

$$\begin{cases} r = \sigma \sum_{i=1}^{k} \vec{R}_y(x_i) \\ \\ s = \sum_{i=1}^{k} \vec{R}_y(x_i) - r \end{cases} \tag{10}$$

With the equivalence of Theorem 1 it is possible to analyse trust networks based on both trust relationships and reputation scores. Fig.4 illustrates a scenario where agent A needs to derive a measure of trust in agent F.

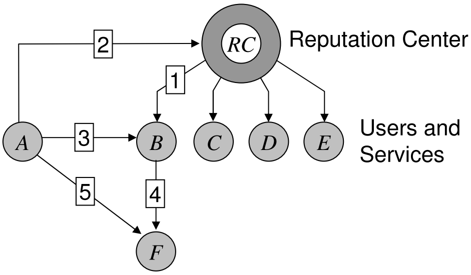


Fig. 4. Combining trust and reputation

Agent B has reputation score $\vec{R}_B^{RC}$ (arrow 1), and agent A has trust ω_{RC}^{A} in the Reputation Centre (arrow 2), so that A can derive a measure of trust in B (arrow 3). Agent B's trust in F (arrow 4) can be recommended to A so that A can derive a measure of trust in F (arrow 5). Mathematically this can be expressed as:

$$\omega_F^A = \omega_{RC}^A \otimes \vec{R}_B^{RC} \otimes \omega_F^B \tag{11}$$

The compatibility between Bayesian reputation systems and subjective logic makes this a very flexible framework for analysing trust in a network consisting of both reputation scores and private trust values.

5 Trust Derivation Based on Trust Comparisons

It is possible that different agents have different trust in the same entity, which intuitively could affect the mutual trust between the two agents. Fig.5 illustrates a scenario where A's trust ω_B^A (arrow 1) conflicts with B's reputation score $\vec{R}_B^{RC}$ (arrow 2).

As a result A will derive a reduced trust value in the Reputation Centre (arrow 3). Assume that A needs to derive a trust value in E, then the reduced trust value must be taken into account when using RC's reputation score for computing trust in E. The operator for deriving trust based on trust conflict produces a binomial opinion over the binary state space $\{x, \overline{x}\}$, where x is a proposition that can be interpreted as x: *"RC provides reliable reputation scores"*, and $\overline{x}$ is its complement. Binomial opinions have the special notation $\omega_x = (b, d, u, a)$ where d represents disbelief in proposition x.

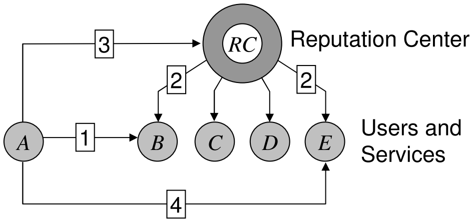
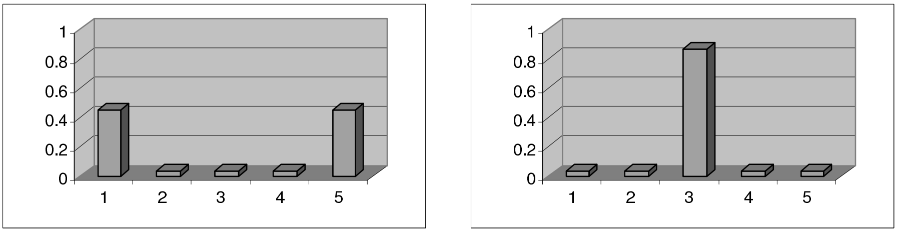


Fig. 5. Deriving trust from conflicting trust

What represents difference in trust values depends on the semantics of the state space. Assume that the state space consists of five rating levels, then Fig.6.a represents a case of polarised ratings, whereas Fig.6.b represents a case of average ratings. Interestingly they have the same point estimate of 0.5 when computed with Eq.(4).



(a) Reputation score from polarized ratings

(b) Reputation score from average ratings

Fig. 6. Comparison of polarized and average reputation scores

We will define an operator which derives trust based on point estimates as defined by Eq.(4). Two agents having similar point estimates about the same agent or proposition should induce mutual trust, and dissimilar point estimates should induce mutual distrust.

Definition 1 (Trust Derivation Based on Trust Comparison)
Let ω^A_B and ω^{RC}_B be two opinions on the same state space B with a set rating levels. A's trust in RC based on the similarity between their opinions is defined as:

$$\omega^A_{RC} = \omega^A_B \downarrow \omega^{RC}_B \quad \textit{where} \begin{cases} d^A_{RC} &= |\sigma(\vec{R}^A_B) - \sigma(\vec{R}^{RC}_B)| \\ u^A_{RC} &= \mathrm{Max}[u^A_B, u^{RC}_B] \\ b^A_{RC} &= 1 - b^A_{RC} - u^A_{RC} \end{cases} \tag{12}$$

The interpretation of this operator is that disbelief in RC is proportional to the greatest difference in point estimates between the two opinions. Also, the uncertainty is equal to the greatest uncertainty of the two opinions.

With the trust comparison trust derivation operator, A is able to derive trust in RC (arrow 3). With the above described trust and reputation measures, A is able to derive trust in E expressed as:

$$\omega^A_E = \omega^A_{RC} \otimes \omega^{RC}_E \tag{13}$$

This provides a method for making trust derivation more robust against unreliable or deceptive reputation scores and trust recommendations.

6 Numerical Example

By considering the scenario of Fig.5, assume that RC has received 5 mediocre and 5 excellent ratings about B as in Fig.6.a, and that A has had 10 average private experiences with B, as in Fig.6.b. Then $\sigma(\vec{R}_B^{RC}) = \sigma(\vec{R}_B^{A}) = 0.5$, so that $d_{RC}^A = 0$. According to Eq.(8) we get $u_B^{RC} = u_B^A = 1/6$, so that $u_{RC}^A = 1/6$, and according to Eq.(12) we get $b_{RC}^A = 5/6$. With $a_{RC}^A = 0.9$ the derived binomial opinion is $\omega_{RC}^A = (5/6, 0, 1/6, 0.9)$, which indicates a relatively strong, but somewhat uncertain trust.

Assume further the aggregate ratings $\vec{R}_E^{RC} = (0, 4, 2, 2, 0)$, i.e. reflecting 0 mediocre, 4 bad, 2 average, 2 good and 0 excellent ratings about E. The base rate vector is set to $\vec{a} = (0.1, 0.2, 0, 2, 0.4, 0.1)$ and the non-informative prior weight $C = 2$. Using Eq.(2), the multinomial reputation score is $\vec{S}_E = (0.02,\ 0.44,\ 0.24,\ 0.28,\ 0.02)$. The point values for each level from mediocre to excellent are: 0.00, 0.25, 0.50, 0.75 and 1.00. Using Eq.(4) the point estimate reputation is $\sigma = 0.46$.

Using Eq.(10) and the fact that $\sum_{i=1}^{k} \vec{R}_E^{RC}(x_i) = 8$, the reputation parameters can be converted to the binomial $(r, s) = (3.68,\ 4.32)$. Using Eq.(8) RC's trust in E in the form of a binomial opinion can be computed as $\omega_E^{RC} = (0.368, 0.432, 0.200, 0.500)$ where the base rate trust has been set to $a_E^{RC} = 0.5$.

The transitivity operator can now be used to derive A's trust in E. The base rate sensitive operator from [11] will be used, which for this example is expressed as:

$$\begin{cases} b_E^{A:RC} = (b_{RC}^A + a_{RC}^A u_{RC}^A) b_E^{RC} \\ d_E^{A:RC} = (b_{RC}^A + a_{RC}^A u_{RC}^A) d_E^{RC} \\ u_E^{A:RC} = 1 - b_E^{A:RC} - d_E^{A:RC} \\ a_E^{A:RC} = a_E^{RC} \end{cases} \tag{14}$$

A's trust in E can then be computed as the opinion $\omega_E^A = (0.362, 0.425, 0.213, 0.500)$, which in terms of probability expectation value is $\mathrm{E}(\omega_E^A) = 0.4686$. This rather weak trust was to be expected from the relatively negative ratings about E.

7 Conclusion

Trust and reputation management represents an important approach for stabilising and moderating online markets and communities. Integration of different systems would be problematic with incompatible trust and reputation systems. We have described how it is possible to elegantly integrate Bayesian reputation systems and trust analysis based on subjective logic. This provides a flexible and powerful framework for online trust and reputation management.

References

1. Jøsang, A.: Artificial reasoning with subjective logic. In: Nayak, A., Pagnucco, M. (eds.) Proceedings of the 2nd Australian Workshop on Commonsense Reasoning, Perth, December 1997. Australian Computer Society (1997)

2. Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
3. Jøsang, A.: Probabilistic Logic Under Uncertainty. In: The Proceedings of Computing: The Australian Theory Symposium (CATS 2007), Ballarat, Australia. CRPIT, vol. 65 (January 2007)
4. Jøsang, A., Gray, E., Kinateder, M.: Simplification and Analysis of Transitive Trust Networks. Web Intelligence and Agent Systems 4(2), 139–161 (2006)
5. Jøsang, A., Hayward, R., Pope, S.: Trust Network Analysis with Subjective Logic. In: Proceedings of the 29th Australasian Computer Science Conference (ACSC 2006), Hobart, Australia. CRPIT, vol. 48 (January 2006)
6. Jøsang, A., Hird, S., Faccer, E.: Simulating the Effect of Reputation Systems on e-Markets. In: Nixon, P., Terzis, S. (eds.) Proceedings of the First International Conference on Trust Management (iTrust), Crete (May 2003)
7. Jøsang, A., Ismail, R.: The Beta Reputation System. In: Proceedings of the 15th Bled Electronic Commerce Conference (June 2002)
8. Jøsang, A., Haller, J.: Dirichlet Reputation Systems. In: The Proceedings of the International Conference on Availability, Reliability and Security (ARES 2007), Vienna, Austria (April 2007)
9. Jøsang, A., Luo, X., Chen, X.: Continuous Ratings in Discrete Bayesian Reputation Systems. In: The Proceedings of the Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM 2008), Trondheim (June 2008)
10. Jøsang, A., Pope, S.: Semantic Constraints for Trust Tansitivity. In: Hartmann, S., Stumptner, M. (eds.) Proceedings of the Asia-Pacific Conference of Conceptual Modelling (APCCM), Newcastle, Australia, February 2005. Conferences in Research and Practice in Information Technology, vol. 43 (2005)
11. Jøsang, A., Pope, S., Marsh, S.: Exploring Different Types of Trust Propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)
12. Withby, A., Jøsang, A., Indulska, J.: Filtering Out Unfair Ratings in Bayesian Reputation Systems. The Icfain Journal of Management Research 4(2), 48–64 (2005)

Controlling Usage in Business Process Workflows through Fine-Grained Security Policies*

Benjamin Aziz[1], Alvaro Arenas[1], Fabio Martinelli[3], Ilaria Matteucci[2,3], and Paolo Mori[3]

[1] STFC Rutherford Appleton Laboratory, Didcot OX11 0QX, UK
{b.aziz,a.e.arenas}@rl.ac.uk
[2] CREATE-NET, Trento, Italy
[3] IIT CNR, Pisa, via Moruzzi, 1 - 56125 Pisa, Italy
{fabio.martinelli,ilaria.matteucci,paolo.mori}@iit.cnr.it

Abstract. We propose a language for expressing fine-grained security policies for controlling orchestrated business processes modelled as a BPEL workflow. Our policies are expressed as a process algebra that permits a BPEL activity, denies it or force-terminates it. The outcome is evaluates with compensation contexts. Finally, we give an example of these policies in a distributed map processing scenario such that the policies constrain service interactions in the workflow according to the security requirements of each entity participating in the workflow.

Keywords: Business Processes, Fine-grained Security Policies, Workflow Monitoring.

1 Introduction

In the last few years, service oriented architectures are gaining a momentum. The automated composition of basic *Web Services* is one of the most promising ideas. Services composition can be made, on the one hand, by a single peer service, which could interact with different systems at different times, preserving their compositionality (*Orchestration*), and on the other hand, it is fundamental to guarantee overall systems functionalities (*Choreography*).

Security is a very important matter in the composition of Web Services. Indeed, services are provided by different entities in the network that could implement different security mechanisms and apply different security policies. The overall interaction of these policies could not allow the correct service workflow execution due to unexpected conflicts among policies. Indeed, services are composed for adhering to a business workflow and access and usage control mechanisms must take into account this view.

In this paper, we mainly focus on fine-grained control of service workflow. In particular, we propose a framework for monitoring the execution of service workflows based

* This work is partially funded by the EU GridTrust project, contract No. 033827 - http://www.gridtrust.eu, and the EU Consequence project, contract No. 214859 - http://www.consequence-project.eu/

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 100–117, 2008.

on policies derived from process description languages similar to those used for defining workflows themselves. These kinds of policy languages are based on a limited set of operators that allow to express basic facts on security relevant actions as well as complex execution patterns. There are operators for describing the sequence of security relevant actions allowed, predicates on the current status of the computation, and the composition of different policies (both conjunction and disjunction). As a matter of fact, by using a process-like policy languages we are able to naturally model history dependent access and usage policies.

To explain more the details of the approach, let us consider that network services are combined by an orchestrator process that, by managing them, satisfies a certain user requirement. Here we define a language that can be used to express fine-grain usage control policies for BPEL-based workflows. In particular it controls access rights and the right to perform any actions in the workflow in general, with particular attention to the BPEL basic activity.

The framework we are going to propose is very general, although here we mainly advocate it for orchestration. In this case, all services are agnostic with respect to the behavior of the other services and the only communication is between the service and the orchestrator. Hence this is a central point of control (and possible failure) of the system and allows for the storage of system relevant information in a natural way. For that reason, we define policies on the orchestrator in order to control essentially its activities and thus the resulting activities of the composed service.

It is also possible to consider that each orchestrated service has a local policy that has to be enforced. In this case each service has a usage control policy defined on it.

The rest of the paper is structured as follows. Section 2 introduces BPEL in an abstract syntax and defines its labeled transition semantics. Section 3 describes our view of the usage control framework for orchestrated services, defines the policy languages and presents the formal semantics of the interaction between the policy and the controlled BPEL workflow. Section 4 shows an example of the applicability of our policies to the domain of distributed map processing. Finally, Section 5 compares our work with related work.

2 BPEL Overview

The Business Execution Language for Web Services (BPEL4WS, simply called here BPEL) [3,4] is a standard specification language for expressing Web service workflows that was adopted by the Organization for the Advancement of Structured Information Standards (OASIS[1]). BPEL resulted from the merge of two earlier workflow languages: XLANG, which is a block structured language designed by Microsoft, and WSFL, which is a graph-based language designed by IBM, and in fact, it adopted their approach in using Web-based interfaces (WSDL, SOAP) as its external communication mechanism while using XML as its specification language. BPEL supports both cases of service composition: service orchestration and service choreography. In the former case, a central process coordinates the execution of a workflow by calling individual services. The services themselves are agnostic to the existence of the workflow. Therefore, the central process acts as the orchestrator of the workflow. In the latter, there is

[1] www.oasis-open.org

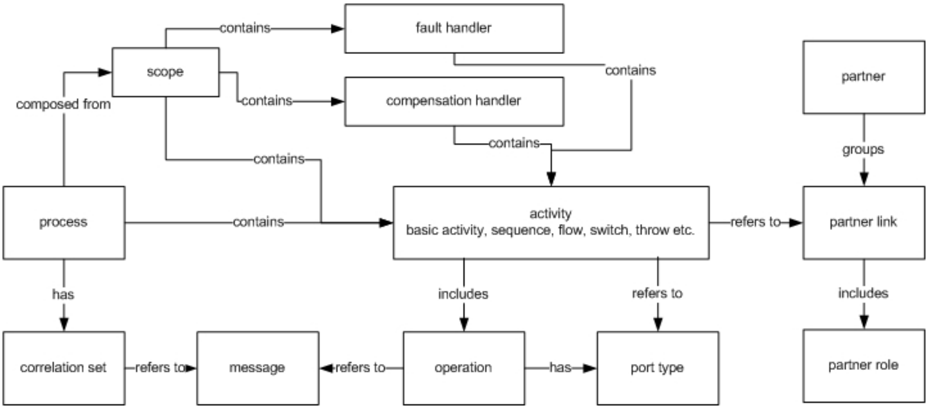


Fig. 1. The BPEL Metamodel

no central coordinator and each service knows its own share of the workflow, in other words, it knows the exact operations it is meant to execute and which other services it should invoke. In this sense, services here are locked in a choreography.

The concept of *executable process* allows for services to be orchestrated in BPEL. On the other hand, the concept of *abstract business protocol* allows for the description of public communication messages without describing the internal details of the workflow process and hence facilitating the choreography of services. In the rest of the paper, we concentrate on the orchestration paradigm since it is more natural to BPEL. There are extensions of BPEL, such as BPEL4Chor [11], that promote the use of BPEL as a choreography language.

Figure 1 depicts the BPEL metamodel. Based to this model, the lifecycle of a BPEL-based executable workflow process is described intuitively as follows. The *process* representing the workflow is invoked by another external process (usually called the client) in which case the workflow process is started within its execution environment, typically a BPEL execution engine. The workflow process contains a description of *activities* that it must perform during the workflow. These activities may be either basic, such as the invocation of Web services, receiving invocations from other services/processes, replying to invocations etc., or structured, which describe the flow of control of basic activities, for example, the sequential composition, parallel composition or the conditional composition of activities. In each basic activity, the name of the *port type*, the name of the *partner link* offering that port type and the name of the *operation* on the port type are specified. Additionally, *parter links* may be grouped as one *partner* and they may have *partner roles*. A process may also have a *correlation set*, which is a set of properties shared by all *messages* in a group of operations offered by a service. A process is divided into *scopes*, each of which contains an activity, a fault handler, a compensation handler and an event handler (we shall ignore event handlers from now on). Fault handlers catch faults and may sometimes re-throw them, whereas compensation handlers of successfully completed activities are used to reverse the effect of those activities (rollback) whenever a fault is caught in the workflow later on.

B	$::=$		activity
		A	basic activity
	$\mid$	skip	do nothing
	$\mid$	throw	fault
	$\mid$	$\mathit{sequence}(B_1,B_2)$	sequential composition
	$\mid$	$\mathit{flow}(B_1,B_2)$	parallel composition
	$\mid$	$\mathit{switch}(\langle \mathit{case}\ b_1 : B_1\rangle,\ldots,\langle \mathit{case}\ b_n : B_n\rangle,\langle \mathit{otherwise}\ B\rangle)$	conditional composition
	$\mid$	$\mathit{scope}\ n : (B,C,F)$	named scope
C,F	$::=$		compensation, fault handler
		$\mathit{compensate}$	compensate-all
	$\mid$	B	activity
P	$::=$	$\{\mid B,F\mid\}$	business process

Fig. 2. Abstract Syntax of the BPEL Language

2.1 BPEL Abstract Syntax

We adopt here an abstract syntax for the BPEL language as defined by [18] and shown in Figure 2.

The syntax defines a BPEL business process as a pair, $\{\mid B, F\mid\}$, consisting of a activity, B, and a fault handler, F. The activity may be composed of several other activities. These could be either a basic activity, A, a do-nothing activity, *skip* or a fault throw activity, *throw*. Examples of basic activities are the communication activities, such as:

- Service invocations in the form of *invoke*$(ptlink, op, ptype)$, in which the operation, op, is invoked belonging to a partner link, $ptlink$, and the operation is invoked on a port type, $ptype$.
- Receiving a request in the form of *receive* $:$ $(ptlink, op, ptype)$, where a service receives a request for an operation op on some port type $ptype$ by some client $ptlink$.
- Replying to a request, *reply* $:$ $(ptlink, op, ptype)$, which generates a reply by calling an operation op over a port type $ptype$ belonging to a partner link $ptlink$.

For simplicity, in the abstract syntax of Figure 2 we have abstracted away all these basic activities and represented them by a simple activity, A, without loss of generality.

An activity may also be a *structured* activity. We consider the following structured activities:

- *sequence*(B_1, B_2): this is a structured activity and it represents the sequential composition of two activities, B_1 and B_2. For B_2 to start executing, B_1 must have already terminated.
- *flow*(B_1, B_2): this is a structured activity and it represents the parallel composition of two activities, B_1 and B_2. We do not assume anything here about the concurrency mode of these two activities (whether it is interleaving or non-interleaving).
- *switch*$(\langle \mathit{case}\ b_1 : B_1\rangle, \ldots, \langle \mathit{case}\ b_n : B_n\rangle, \langle \mathit{otherwise}\ B\rangle)$: this activity represents the conditional case-based statement, where an activity B_i is chosen if its logical condition, b_i, is true. If there are more than one logical conditions that are true, then one of these is chosen non-deterministically. Otherwise, the default B is executed if none of the logical conditions is satisfied. Conditions b are assumed to be expressed in some form of first order logic.

- *scope* n : (B, C, F): this is a scope named n, which has a default activity, B, a compensation handler, C and a fault handler F. The scope usually runs as the default activity, B. If this executes successfully, the compensation handler, C, is installed in the context. Otherwise, the fault handler, F, is executed.

Fault and compensation handlers have the same definition as activities except that they can perform compensation-all calls. For simplicity, we do not consider named compensations, since these are a special case of compensation-all that require special operations to search for the name of the compensation scope belonging to past finished activities.

2.2 Example: Distributed Map Processing

We consider here a simple example of a distributed map processing application inspired by one of the application scenarios of project GridTrust [13]. The workflow representing interactions among the different components of the application are illustrated in Figure 3.

The application consists of a main orchestrator process, which is the *server farm*, that interacts with a couple of services, the *processing centre* and the *storage resources* services, whenever the server farm receives a request from the *client*. The workflow proceeds as follows:

- A client cartographer submits a request to the server farm process, which advertises a map processing service that can create new maps. The request contains any relevant information related to the old and new maps requested by the client. As an example, we consider that the compensation for receiving the client's request is to request back to the client to send the map job again.
- The server farm process invokes a local or a network-based resource storage service and stores on that service data related to the job submitted by the client. We consider that this invocation will be compensated by deleting the job data from the storage service.
- The server farm process next submits a map processing request to a processing centre service requesting, which then retrieves information relevant to the new map and then sends the results back to the server farm.
- Once the processing centre has ensured that the server farm is authorized to modify the map, the processing centre processes the job request and sends the results back to the server farm. These results contain the new map. We consider here that if the server farm is unable to receive the results of the map processing, then it will ask for a compensation of the finished previous activities.
- After having received the results from the processing centre, the server farm carries on final customisation processing on the new map and once finished, sends back the result to the client cartographer.
- The client cartographer now is expected to make a payment to (possibly as a result of an off-line invoice it received) the server farm process. This payment is received and checked by the server farm process. If ok, the client is acknowledged.

The basic BPEL definition of the main server farm process is shown in Figure 4, where we have used the syntactic sugar *sequence*$(B_1, \ldots, B_n)$ instead of *sequence*$(B_1,$ *sequence*$(\ldots, B_n))$.

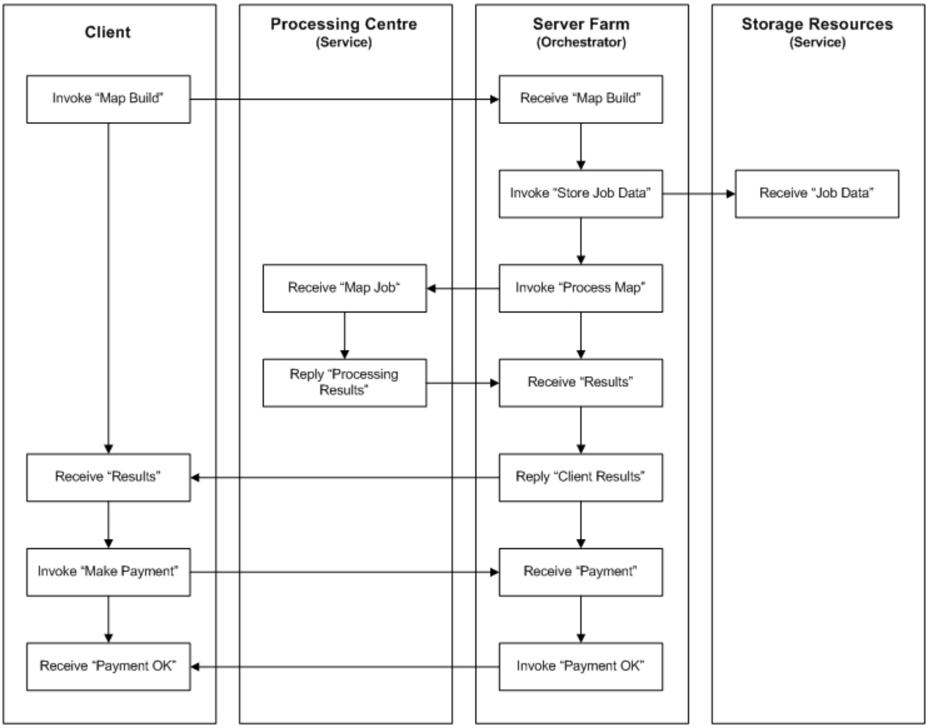


Fig. 3. Workflow for the Distributed Map Processing Application

$$
\begin{aligned}
\mathit{ServerFarm} = \{|\mathit{sequence}(&\\
&\mathit{scope\ req} : (\mathit{receive}(\mathit{Client}, \mathit{mapBuild}, \mathit{Map\ Build\ Port}), C_{req}, \mathit{throw}),\\
&\mathit{scope\ str} : (\mathit{invoke}(\mathit{Storage\ Resources}, \mathit{storeJobData}, \mathit{Resource\ Port}), C_{str}, \mathit{throw}),\\
&\mathit{scope\ prcinv} : (\mathit{invoke}(\mathit{Processing\ centre}, \mathit{processMap}, \mathit{Process\ Port}), \mathit{skip}, \mathit{throw}),\\
&\mathit{scope\ prcrec} : (\mathit{receive}(\mathit{Processing\ centre}, \mathit{inputProcessingResults}, \mathit{Process\ Results\ Port}),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{skip}, \mathit{compensate}),\\
&\mathit{scope\ res} : (\mathit{reply}(\mathit{Client}, \mathit{mapResults}, \mathit{Map\ Results\ Port}), \mathit{skip}, \mathit{throw}),\\
&\mathit{scope\ pay} : (\mathit{receive}(\mathit{Client}, \mathit{makePayment}, \mathit{Payment\ Port}), \mathit{skip}, \mathit{throw}),\\
&\mathit{scope\ ack} : (\mathit{invoke}(\mathit{Client}, \mathit{allOK}, \mathit{Payment}), \mathit{skip}, \mathit{throw})),\\
&\mathit{compensate}|\}
\end{aligned}
$$

where,

$$
\begin{aligned}
C_{req} = \quad & \mathit{sequence}(\mathit{invoke}(\mathit{Client}, \mathit{resendMap},\\
&\qquad \mathit{Map\ Build\ Port}), \mathit{receive}(\mathit{Client}, \mathit{mapBuild}, \mathit{Map\ Build\ Port}))
\end{aligned}
$$

and,

$$C_{str} = \quad \mathit{invoke}(\mathit{Storage\ Resources}, \mathit{deleteJobData}, \mathit{Resource\ Port})$$

Fig. 4. The Server Farm Process

3 Fine-Grained Policies for Orchestration

In this section we propose our architecture for controlling the right to perform BPEL basic activities in the workflow. As a matter of fact, we consider a network of services and a user's request input of an orchestrator process that guarantees that the request is satisfied by managing the given set of services.

We focus our attention on fine-grained control of services workflow. Indeed, in the next section, we will define a language that can be used to express fine-grain usage control policies for BPEL-based workflow.

We suppose that all services are agnostic with respect to the behavior of the other services and the only communication is between each service and the orchestrator. Hence the orchestrator is a central point of control of the system and it allows for the storage of system relevant information in a natural way. In [14] we proposed a semantics definition for describing a possible behavior of an orchestrator process by modeling the web services scenario by process algebra. Here we consider to already have the specification of the orchestrator process and we provide a method to control the workflow. For that reason, we define usage control policies for the orchestrator process in such a way it is possible to control essentially its basic activities.

We also consider that each orchestrated service has a local policy that has to be enforced, for instance, it could be possible that a service requires to interact only with a specified orchestrator and it decides to ignore other external requests. Moreover it is possible to consider each service as an active part of the architecture, for instance, in the *receive/response* activity, when the service responds to an orchestrator invocation it acts as a subject. The architecture we propose is represented in Figure 5. The Orchestrator receives a request from the user and interacts with services by invoking them through the *invoke* basic activity, and by receiving the results through the *receive* basic activity.

The Orchestrator Controller (OC) is integrated in the orchestrator. In this way, the OC is able of intercepting every basic activity of orchestrator, before that they are actually performed. The OC suspends the intercepted activity and it checks whether it is allowed by the security policy in the current state. Each service has a Usage Control

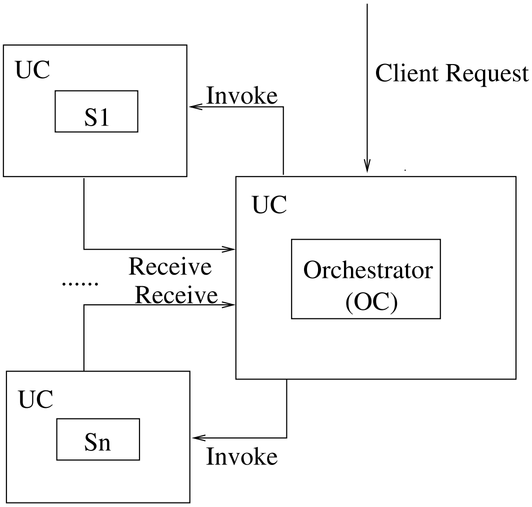


Fig. 5. Architecture with localized controllers

component (UC), that at leat includes a Policy Decision Point (PDP), that is the component that performs the decision activity, plus other components for service/contextual information or attributes management. If the activity is allowed by the PDP, the OC resumes it, otherwise the activity is skipped. It is worthwhile noticing that with orchestrators that mediates all the service invocations, the UC is able to gain information from the UC of the services. Being this already a point of centralization of the service architecture, eventual synchronizations among different UC (e.g., for the exchange of credentials).woudl not create additional architectural dependencies.

3.1 The Policy Specification Language

The policy specification language that we propose here to express fine-grained security policies for BPEL-based business processes is based on the POLicy-based Process Algebra (POLPA) introduced in [1]. Being obtained from a formal language for describing concurrent processes, POLPA is able to naturally describe correct workflow executions in a simple way. For instance we may naturally describe allowed sequence of actions as well as temporal dependencies from set of actions. We have also a simply rule for composing policies that must hold in parallel.

Figure 6 illustrates the abstract syntax of our POLPA language. The syntax is described informally as follows:

- Policies, *Pol*: These consist of the deny-all policy, $\bot$, which prohibits any BPEL activities from taking place and the allow-all policy, $\top$, which allows any BPEL activities to take place. The $A.Pol$ policy permits a basic BPEL activity, A, then continues as *Pol*. $\phi.Pol$ continues as *Pol* if the predicate ϕ is equivalent to true. Finally, Pol_1 *or* Pol_2 is the non-deterministic choice between Pol_1 and Pol_2, and Pol_1 *par* Pol_2 represents the parallel composition between Pol_1 and Pol_2.
- Predicates, ϕ: These are logical operators on usage patterns that are evaluated on the current local states of the business process orchestrator and any of the services involved in the business workflow. Such local states could include the authorization control state (such as ACLs) and history of all the activities run so far (successfully or unsuccessfully) and the time of their execution. Examples of usage patterns include:
  - ○ *Authorization*: These predicates determine whether a subject (the entity executing the business process or the service) will be authorized to perform an activity

$$
\begin{array}{lcll}
Pol ::= & & & \text{policy} \\
 & & \bot & \text{deny-all} \\
 & | & \top & \text{allow-all} \\
 & | & A.Pol & \text{action sequential composition} \\
 & | & \phi.Pol & \text{predicate sequential composition} \\
 & | & Pol_1 \ \mathit{or} \ Pol_2 & \text{non-deterministic choice} \\
 & | & Pol_1 \ \mathit{par} \ Pol_2 & \text{parallel composition}
\end{array}
$$

Fig. 6. The POLPA Policy Specification Language

or not. The model of authorization could be that of Role-Based Access Control (RBAC) [19], which can be modeled as the following pair of predicates:

$$isAssigned : Sbj \times Role \rightarrow \mathbb{B} \tag{1}$$

$$canPerform : Role \times B \rightarrow \mathbb{B} \tag{2}$$

The first predicate denotes that a subject, *Sbj*, is assigned to a role, *Role* whereas the second predicate denotes that a role, *Role*, can perform an activity, B. Similar predicates might be used for Role-based Trust Management (RTML) extended with weights for quantitative notions of trust as done in [9].

- *Reputation*: This predicate collects the reputation of a subject (the entity executing the business process or the service). The reputation is a number ranging from 0 (un-trusted user) to 1 (trusted user).

$$Rep : Sbj \rightarrow [0, 1] \tag{3}$$

We now define a transition semantics for the policy language in terms of a labeled reduction relation, $\xrightarrow{\mu}$, as shown in Figure 7, where $\mu \in \{A, \tau\}$, and τ is a silent transition. The first rule, $(POL1)$, says that a policy $A.P$ guarded by an action A can perform a transition to remove that guard and continue as the residue P. We consider here actions to be the same as the BPEL basic activities (As in Figure 2). The transition will emit A to an external observer. Rule $(POL2)$ states that a true predicate can allow a policy guarded by that predicate to continue as the residue emitting the silent action τ. Rules $(POL3)$ and $(POL4)$ deal with the case of non-deterministic choice between two policies whereas rules $(POL5)$–$(POL7)$ deal with the case of parallel composition of policies.

For example, given that A_1 and A_2 represent basic activities and ϕ_1 and ϕ_2 are predicates, the following policy:

$\phi_1.A_1.\phi_2.A_2$

allows first the execution of the activity A_1, if the conditions represented by the predicate ϕ_1 are satisfied, followed by the activity A_2, if the conditions represented by the predicate ϕ_2 are satisfied.

$$\begin{array}{ll}
(POL1) & \mathit{A.Pol} \xrightarrow{A} \mathit{Pol} \\
(POL2) & \phi \text{is True} \;\Rightarrow\; \phi.\mathit{Pol} \xrightarrow{\tau} \mathit{Pol} \\
(POL3) & \mathit{Pol}_1 \xrightarrow{\mu} \mathit{Pol}'_1 \;\Rightarrow\; \mathit{Pol}_1 \; \mathit{or} \; \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_1 \\
(POL4) & \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_2 \;\Rightarrow\; \mathit{Pol}_1 \; \mathit{or} \; \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_2 \\
(POL5) & \mathit{Pol}_1 \xrightarrow{\mu} \mathit{Pol}'_1 \;\Rightarrow\; \mathit{Pol}_1 \; \mathit{par} \; \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_1 \; \mathit{par} \; \mathit{Pol}_2 \\
(POL6) & \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_2 \;\Rightarrow\; \mathit{Pol}_1 \; \mathit{par} \; \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}_1 \; \mathit{par} \; \mathit{Pol}'_2 \\
(POL7) & \mathit{Pol}_1 \xrightarrow{\mu} \mathit{Pol}'_1, \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_2 \;\Rightarrow\; \mathit{Pol}_1 \; \mathit{par} \; \mathit{Pol}_2 \xrightarrow{\mu} \mathit{Pol}'_1 \; \mathit{par} \; \mathit{Pol}'_2
\end{array}$$

Fig. 7. Labelled Transition Semantics for POLPA

Predicates can express conditions on the parameters of the activity or on the current state. For example, the following policy:

(*Rep*(Processing Centre)≥ 0.7).
invoke(*Processing Centre*,*processMap*,*Process Port*)

allows the execution of the activity *invoke*(*Processing Centre*, *processMap*, *Process Port*) only if the reputation of the service *Processing Centre* is greater than 0.7.

3.2 POLPA-Controlled Semantics for BPEL

In this section, we show how POLPA policies can be used to control the behaviour of BPEL business processes. First, we define the concept of a *decision* that a policy enforces on activities. This decision, in general, is represented by the box symbol, $\Box$, which could stand for any of the following decisions:

- $\boxdot$: the security policy decides to permit the activity to execute and succeed.
- $\boxtimes$: the security policy decides to deny the permission to execute the activity.
- $\overline{\boxtimes}$: the activity was forced to terminate prematurely (perhaps by the security policy enforcement point or the BPEL engine).

Our policy decisions are specializations of the termination points (successful, failed and forced) as introduced by [18].

Our semantics is defined using a big-step transition relation in the form of:

$$\mathit{Pol} \vdash B, \alpha \longrightarrow \Box, \beta \tag{4}$$

which states that an activity B running with the compensation context α under the security policy *Pol* will be allowed to terminate (either by permitting, denying or force-terminating it) resulting in the compensation context β.

Compensation contexts are ranged over by α, β, γ, and these are defined as sequences of *compensation closures*. A compensation closure, $(n : C : \alpha)$, denotes that the compensation handler activity C was installed in a scope named n and if run, it will do so in the compensation context, α. Appendix A presents the full description of the POLPA-controlled semantics rules for BPEL language.

4 Example Revisited

In what follows, we define POLPA policies for the main server farm process as well as for the client, the processing centre service and the storage resources service. We start with the policy for the server farm process shown in Figure 8.

Each line corresponds to one of the activities that the server farm is expected to perform in the distributed map processing workflow. The activities are clearly all allowed by the policy, which then terminates with a deny-all residue, $\perp$. The interesting parts of the policy are the different predicates accompanying each activity. These are described as follows. After receiving the client's request, the policy checks whether the client is in

$$\begin{aligned}
\mathit{Pol}_{SF} = {} & \mathit{receive}(\mathit{Client}, \mathit{mapBuild}, \mathit{Map\ Build\ Port}).\\
& (\mathit{isAssigned}(\mathit{Client}, \mathit{Cartographer})\ \wedge\\
& \mathit{canPerform}(\mathit{Cartographer}, \mathit{Server\ Farm}, \mathit{mapBuild}, \mathit{Map\ Build\ Port})).\\
& (\mathit{securityLevel}(\mathit{Job\ Data}) \leq \mathit{securityLevel}(\mathit{Storage\ Resource})).\\
& \mathit{invoke}(\mathit{Storage\ Resources}, \mathit{storeJobData}, \mathit{Resource\ Port}).\\
& (\mathit{Rep}(ProcessingCentre) \geq 0.7).\\
& \mathit{invoke}(\mathit{Processing\ Centre}, \mathit{processMap}, \mathit{Process\ Port}).\\
& \mathit{receive}(\mathit{Processing\ Centre}, \mathit{inputProcessingResults}, \mathit{Process\ Results\ Port}).\\
& (\mathit{age}(\mathit{Processing\ Data}) \leq 12\ \mathit{Days}).\\
& (\mathit{Current\ System\ Date\ And\ Time} \leq \mathit{Deadline\ Date\ And\ Time}).\\
& \mathit{invoke}(\mathit{Client}, \mathit{mapResults}, \mathit{Map\ Results\ Port}).\\
& \mathit{receive}(\mathit{Client}, \mathit{makePayment}, \mathit{Payment\ Port}).\\
& (\mathit{Client\ Payment} = \mathit{Expected\ Payment}).\\
& \mathit{invoke}(\mathit{Client}, \mathit{allOK}, \mathit{Payment}).\bot
\end{aligned}$$

Fig. 8. The POLPA policy for the Server Farm process

$$\begin{aligned}
\mathit{Pol}_{PC} = {} & \mathit{receive}(\mathit{Server\ Farm}, \mathit{processMap}, \mathit{Process\ Port}).\\
& (\mathit{isAssigned}(\mathit{Server\ Farm}, \mathit{Authorised\ Reader}) \wedge\\
& \mathit{canPerform}(\mathit{Authorised\ Reader}, \mathit{Processing\ centre}, \mathit{processMap}, \mathit{Process\ Port})).\\
& (\mathit{securityLevel}(\mathit{TxData}(\mathit{Processing\ centre})) \leq \mathit{securityLevel}(\mathit{Server\ Farm})).\\
& \mathit{reply}(\mathit{Server\ Farm}, \mathit{inputProcessingResults}, \mathit{Process\ Results\ Port}).\bot
\end{aligned}$$

Fig. 9. The POLPA policy for the Processing Centre Service

fact assigned a cartographer's role and whether that role is permitted to invoke the *mapBuild* operation on port *Map Build Port*. If this is the case, the policy continues to the next activity. Here, the policy checks whether the security level of the received job data is lower or equal to the security level of the resources storage service the server farm is planning to store the data on. If so, the policy allows the storage to occur and continues to the next activity. Here, a predicate on the processing centre's reputation being a minimum of 0.7 is checked. If true, the processing centre's *processMap* operation is permitted. Then the policy allows for the results of the map processing to received, after which the freshness of the new map is checked. This is necessary since information on the map, such as locations of petrol stations, restaurants and even the topography of roads may have changed if the map is older than 12 days.

Once the results are ready to be sent to the client, the policy makes sure that the current time and date are within the deadline for the map job request agreed with the client. This then allows for the results to be sent back to the client. The policy then permits the receipt of the payment for the job after which it checks with a predicate whether the payment was the expected amount. If so, the policy allows for an extra activity which invokes the *allOK* on the client. Once this is done, the policy leaves a residue of a deny-all policy, $\bot$.

The POLPA policy for the processing centre is shown in Figure 9. In this policy, the processing centre is permitted to receive a request for a map analysis job from the server farm. Once this request is received, the policy checks whether the server farm is assigned the role of an authorised reader and whether an authorised reader is permitted to invoke the *processMap* operation. If this is the case, the policy then checks a predicate on the security level of the data that will be transmitted to back to the server farm and whether this security level is lower or equal to the server farms's level. If this is the case, it will permit a reply to the server farm with the new map data and ends in a deny-all policy.

Next, we define the POLPA policy for the storage resources service as shown in Figure 10.

$$\begin{aligned} Pol_{SR} = \quad & \mathit{receive}(\mathit{Server\ Farm}, \mathit{storeJobData}, \mathit{Resource\ Port}). \\ & \neg(\mathit{dataSize}(\mathit{Job\ Data}) \leq 1\ \mathit{GB}).\bot \end{aligned}$$

Fig. 10. The POLPA policy for the Storage Resources Service

This policy is simple; it basically ensures that any data written to the resources must not exceed the size of 1 GB per job. Finally, we define the POLPA policy for the client as in Figure 11. The policy for the client ensures that the reputation level of the server farm is indeed higher than the minimum of 0.9 required by the client. After that, the client expects to receive the results of its map processing job. These results are checked for their minimum quality as indicated by a set of quality criteria, *QualityCriteria*. Finally, the client makes a payment according to some invoice bill it received offline from the server farm and then expects to receive the payment acknowledgement from the server farm. Once this is done, the policy then moves to the deny-all policy.

$$\begin{aligned} Pol_{cl} = \quad & (\mathit{Rep}(\mathit{Server\ Farm}) \geq 0.9). \\ & \mathit{invoke}(\mathit{Server\ Farm}, \mathit{mapBuild}, \mathit{Map\ Build\ Port})). \\ & \mathit{receive}(\mathit{Server\ Farm}, \mathit{mapResults}, \mathit{Map\ Results\ Port}). \\ & (\mathit{Quality}(\mathit{Map\ Results}) \subseteq \mathit{QualityCriteria}). \\ & (\mathit{Client\ Payment} = \mathit{Invoice\ Amount}). \\ & \mathit{invoke}(\mathit{Server\ Farm}, \mathit{makePayment}, \mathit{Payment\ Port}). \\ & \mathit{receive}(\mathit{Server\ Farm}, \mathit{paymentAck}, \mathit{Payment\ Port}).\bot \end{aligned}$$

Fig. 11. The POLPA policy for the Client

5 Related Work

In literature there are several works [12,16,20,8] about usage control or access control applied to GRID/web services in order to guarantee secure access to those services. Most of these are concerned with the so-called coarse grain service level, especially for GRIDs. Less work has been performed on fine-grain authorization and access control at workflow level as we are advocating here.

For instance, [1] proposes the adoption of a fine-grained authorization system to enhance the security of Grid Computational Services. As a matter of fact, Grid Computational Services are shared in the Grid environment and execute unknown applications on behalf of potentially unknown users. In this framework an application executed on the Grid Computational Service on behalf of a remote grid user is not viewed as an atomic entity, as in standard authorization systems that, once the right of executing the application has been granted does not perform any further control on what the application does. Instead, the proposed framework takes into account the behavior of the application, and it monitors all the actions that the application tries to perform on the underlying resource and enforces a fine-grained policy that defines the allowed behaviors for the applications. The application was written in Java while here we focus on BPEL specifications (thus we describe the business logic of the services).

[2] describes an inline approach to monitor the services. The authors presented an approach to specify monitoring directives, and weave dynamically into the process they belong to and a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time. Moreover it presents a user-oriented language to integrate data acquisition and analysis into monitoring rules. Our approach permits us to control all the workflow activities by monitoring, on one side, the orchestrator process, *i.e.*, by monitoring, for instance, the *invoke* actions, and, on the other side, the services by checking, for instance, *receive/response* activities. As a matter of fact we are able to enforce global policies on the orchestrator process and local policies on the service side.

In [21] the authors proposed a usage control (UCON) based authorization framework for collaborative application. They described their theory in particular for heterogeneous distribution of resources and the various modes of collaborations that exist between users, virtual organizations, and resource providers. In our paper we propose how fine-grained usage control can be used also in the field of web services. As a matter of fact we use fine-grained access control to protect both services and orchestrator in order to have a secure workflow.

[15] concerns with the access control for BPEL based processes. In particular the authors presents an approach to integrate Role-Based Access Control (RBAC) and BPEL on the meta-model level. They describe a mapping of BPEL to RBAC elements and extracts them from BPEL. In particular they presents a XSLT script which transforms BPEL processes to RBAC models in an XML format. On the contrary [5] presents two languages, RBAC-WS-BPEL and BPCL in order to be able to specify authorization information associating users with activities in the business process and authorization constraints on the execution of activities. The focus is mainly on RBAC models although with the introduction of BPCL, the authors recognize the need for languages for expressing constraints on activities. BPCL seems able to model sequences, but it seems difficult to model more complex patterns. Instead, POLPA language was exactly derived from process description languages able to naturally express a significant variety of patterns.

In [10], the author presents an analysis of the satisfiability of task-based workflows. The model of workflows adopted is enriched with entailment constraints that can be used for expressing cardinality and separation of duties requirements. Given such and other authorization constraints, the analysis then answers questions related to whether the workflow can or cannot be achieved and whether an enforcement point can or cannot be designed based on all instances of the workflow. Our work is much more focussed on the enforcement of usage control rather than performing analysis that is actually part of our future work.

In [7], the author shows how basic features of XCAML may be encoded in the CSP process algebra. Our POLPA language inherits some features from such process algebra and thus we think also POLPA is able to encode XCAML policies (at least in a model theoretic point of view). The author also notices as policies for workflows should be based on process description languages rather than simply RBAC ones. We share this view.

[17] presents a general discussion on the authorization and usage control frameworks for SOA. The treatment is very general and several concepts are recalled. They describe several possible architectures for controlling usage of service and our could be considered as an instance. However, their work does not discuss policy languages or BPEL semantics and thus our could be seen as instantiation.

Finally, [6] proposes an algebra for composing access control policies in a modular manner. Their framework yields an implementation in based on logic programming and partial evaluation techniques. However, their application domain is general and does not tie with any particular paradigm, such as our workflow-based business process paradigm.

Acknowledgments. We would like to thank the anonymous reviewers for their helpful comments.

References

1. Baiardi, F., Martinelli, F., Mori, P., Vaccarelli, A.: Improving Grid Services Security with Fine Grain Policies. In: Meersman, R., Tari, Z., Corsaro, A. (eds.) OTM-WS 2004. LNCS, vol. 3292, pp. 123–134. Springer, Heidelberg (2004)
2. Baresi, L., Guinea, S.: Towards Dynamic Monotoring of WS-BPEL Processes. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 269–282. Springer, Heidelberg (2005)
3. BEA, IBM, Microsoft, SAP, and Siebel. Business Process Execution Language for Web Services Version 1.1. Public Specification (2003)
4. BEA, IBM, Microsoft, SAP, and Siebel. Web Services Business Process Execution Language Version 2.0. OASIS Standard (2007)
5. Bertino, E., Crampton, J., Paci, F.: Access Control and Authorization Constraints for WS-BPEL. In: Proceedings of the 2006 IEEE International Conference on Web Services, Chicago, Illinois, USA, pp. 275–284. IEEE Computer Society, Los Alamitos (2006)
6. Bonatti, P.A., di Vimercati, S.D.C., Samarati, P.: A Modular Approach to Composing Access Control Policies. In: Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS 2000), Athens, Greece, November 2000, pp. 164–173. ACM Press, New York (2000)
7. Bryans, J.: Reasoning about xacml policies using csp. In: SWS. ACM Press, New York (2005)
8. Chadwick, D., Otenko, A.: The permis x.509 role based privilege management infrastructure. In: SACMAT 2002: Proceedings of the seventh ACM symposium on Access control models and technologies, pp. 135–140. ACM Press, New York (2002)
9. Colombo, M., Martinelli, F., Mori, P., Petrocchi, M., Vaccarelli, A.: Fine grained access control with trust and reputation management for globus. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1505–1515. Springer, Heidelberg (2007)

10. Crampton, J.: An Algebraic Approach to the Analysis of Constrained Workflow Systems. In: Proceedings of the 3rd Workshop on Foundations of Computer Security, pp. 61–74 (2004)
11. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: Proceedings of the IEEE 2007 International Conference on Web Services (ICWS 2007), Salt Lake City, Utah, USA. IEEE Computer Society, Los Alamitos (2007)
12. Foster, I., Kesselman, C., Pearlman, L., Tuecke, S., Welch, V.: A community authorization service for group collaboration. In: Proceedings of the3rd IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2002), pp. 50–59 (2002)
13. GridTrust. Deliverable D5.1(M19) Specifications of Applications and Test Cases (2007)
14. Martinelli, F., Matteucci, I.: Synthesis of web services orchestrators in a timed setting. In: Dumas, M., Heckel, R. (eds.) WS-FM 2007. LNCS, vol. 4937, pp. 124–138. Springer, Heidelberg (2008)
15. Mendling, J., Strembeck, M., Stermsek, G., Neumann, G.: An Approach to Extract RBAC Models from BPEL4WS Processes. In: Proceedings of the Thirteenth IEEE International Workshops on Enabling Technologies (WETICE 2004): Infrastructure for Collaborative Enterprises, Modena, Italy, pp. 81–86. IEEE Computer Society, Los Alamitos (2004)
16. Pearlman, L., Kesselman, C., Welch, V., Foster, I., Tuecke, S.: The community authorization service: Status and future. In: Proceedings of Computing in High Energy and Nuclear Physics (CHEP 2003): ECONF (2003) C0303241:TUBT003
17. Pretschner, A., Massacci, F., Hilty, M.: Usage control in service-oriented architectures. In: Lambrinoudakis, C., Pernul, G., Tjoa, A.M. (eds.) TrustBus. LNCS, vol. 4657, pp. 83–93. Springer, Heidelberg (2007)
18. Qiu, Z., Wang, S., Pu, G., Zhao, X.: Semantics of BPEL4WS-Like Fault and Compensation Handling. In: Fitzgerald, J.S., Hayes, I.J., Tarlecki, A. (eds.) FM 2005. LNCS, vol. 3582, pp. 350–365. Springer, Heidelberg (2005)
19. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. Computer 29(2), 38–47 (1996)
20. Thompson, M., Essiari, A., Mudumbai, S.: Certificate-based authorization policy in a pki environment. ACM Transactions on Information and System Security (TISSEC) 6(4), 566–588 (2003)
21. Zhang, X., Nakae, M., Covington, M.J., Sandhu, R.: A usage-based authorization framework for collaborative computing systems. In: SACMAT 2006: Proceedings of the eleventh ACM symposium on Access control models and technologies, pp. 180–189. ACM Press, New York (2006)

A Semantics Rules for POLPA-Controlled

The interested reader can find here a detailed semantics definition even if it is not determinant to understand the paper.

We sometimes use the syntactic sugar box, $\boxslash$, to mean either $\boxtimes$ or $\overline{\boxtimes}$, and $\boxminus$ to mean either $\boxdot$ or $\boxtimes$. Additionally, we define the operator, $\otimes$, as follows:

$\otimes$	$\boxdot$	$\overline{\boxtimes}$	$\boxtimes$
$\boxdot$	$\boxdot$	$\overline{\boxtimes}$	$\boxtimes$
$\overline{\boxtimes}$	$\overline{\boxtimes}$	$\overline{\boxtimes}$	$\boxtimes$
$\boxtimes$	$\boxtimes$	$\boxtimes$	$\boxtimes$

This last operator shows that the success of permitting an activity is empowered by the forced termination decision and the latter is empowered by the complete denial for executing the activity. The $\otimes$ operator is needed to expresses policy decisions regarding activities composed in parallel, where the denial of one activity forces the termination of all the other activities in parallel with it, as will be explained later in the semantics.

As we have already said, our semantics is defined using a big-step transition relation in the form of:

$$\mathit{Pol} \vdash B, \alpha \longrightarrow \Box, \beta \qquad (5)$$

Since POLPA has a small-step semantics and the controlled semantics of BPEL is big-step, we need to define the concept of a *residual policy* defined as a relation, $\mathit{res}(\mathit{Pol},B,\Box)$, which is defined as the residue of P at the time the activity B has reached the $\Box$ termination status. This is formally defined as:

$$\mathit{res}(\mathit{Pol},B,\Box) = \mathit{Pol}' \ \text{ such that } \ (\exists \mu_1,\ldots,\mu_n : \mathit{Pol} \xrightarrow{\mu_1} \ldots \xrightarrow{\mu_n} \mathit{Pol}') \Leftrightarrow (\mathit{Pol} \vdash B, \alpha \longrightarrow \Box, \beta) \qquad (6)$$

This means that by the time the B activity has reached $\Box$, the POLPA policy will have performed basic-activity transitions $A_1 \ldots A_n$. If $n = 0$, this implies that $\mathit{Pol} = \mathit{Pol}'$, in other words, the activity terminated without the need for the policy to make any transitions (as the semantics will demonstrate, this could be due to the activity containing only *skip*).

Now, we define the rules for the transition relation $\longrightarrow$ as in Figure 12.

Informally, Rule $(BPEL1)$ assumes that a *skip* activity is always permitted by any policy. Rule $(BPEL2)$ assumes that a fault *throw* resembles the situation where the security policy has denied the execution of the current activity encompassing *throw*. This is true for any security policy. Rules $(BPEL3)$ and $(BPEL4)$ state that a basic activity is permitted (resp. denied) execution by the policy enforcement point if the policy can (resp. cannot) make a transition labeled with that basic activity. The same outcome also can be reached if the policy is the allow-all (resp. deny-all) policy. Rules $(BPEL5)$ and $(BPEL6)$ deal with the case of sequential composition of activities. Rule $(BPEL5)$ states that if the first activity in the composition is permitted by the policy to execute and succeed, then the outcome of the composition is the outcome of the second activity as decided by whatever policy remains from the first activity. Rule $(BPEL6)$ states that if the first activity is denied execution or force-terminated,

$$\begin{array}{lll}
(\mathit{BPEL}1) & \mathit{Pol} \vdash \mathit{skip}, \alpha \longrightarrow \boxdot, \alpha & \\
(\mathit{BPEL}2) & \mathit{Pol} \vdash \mathit{throw}, \alpha \longrightarrow \boxtimes, \alpha & \\[1em]
(\mathit{BPEL}3) & (\exists \mathit{Pol}' : \mathit{Pol} \xrightarrow{A} \mathit{Pol}') \ \vee \ (\mathit{Pol} = \top) \ \Rightarrow \ \mathit{Pol} \vdash A, \alpha \longrightarrow \boxdot, \alpha & \\
(\mathit{BPEL}4) & \neg(\exists \mathit{Pol}' : \mathit{Pol} \xrightarrow{A} \mathit{Pol}') \ \vee \ (\mathit{Pol} = \bot) \ \Rightarrow \ \mathit{Pol} \vdash A, \alpha \longrightarrow \boxtimes, \alpha & \\[1em]
(\mathit{BPEL}5) & \mathit{Pol} \vdash B_1, \alpha \longrightarrow \boxdot, \gamma \ \wedge \ \mathit{res}(\mathit{Pol}, B_1, \boxdot) \vdash B_2, \gamma \longrightarrow \square, \beta \ \Rightarrow & \\
 & \mathit{Pol} \vdash \mathit{sequence}(B_1, B_2), \alpha \longrightarrow \square, \beta & \text{where } \square \in \{\boxdot, \boxtimes, \overline{\boxtimes}\} \\
(\mathit{BPEL}6) & \mathit{Pol} \vdash B_1, \alpha \longrightarrow \boxslash, \gamma \ \Rightarrow \ \mathit{Pol} \vdash \mathit{sequence}(B_1, B_2), \alpha \longrightarrow \boxslash, \gamma & \text{where } \boxslash \in \{\boxtimes, \overline{\boxtimes}\} \\[1em]
(\mathit{BPEL}7) & \exists i \in \{1, \ldots, n\} : b_i = \texttt{true} \ \wedge \ \mathit{Pol} \vdash B_i, \alpha \longrightarrow \square, \gamma \ \Rightarrow & \\
 & \mathit{Pol} \vdash \mathit{switch}(\langle \mathit{case}\ b_1 : B_1 \rangle, \ldots, \langle \mathit{case}\ b_n : B_n \rangle, \langle \mathit{otherwise}\ B \rangle), \alpha \longrightarrow \square, \gamma & \\
(\mathit{BPEL}8) & \forall i \in \{1, \ldots, n\} : b_i = \texttt{false} \ \wedge \ \mathit{Pol} \vdash B, \alpha \longrightarrow \square, \gamma \ \Rightarrow & \\
 & \mathit{Pol} \vdash \mathit{switch}(\langle \mathit{case}\ b_1 : B_1 \rangle, \ldots, \langle \mathit{case}\ b_n : B_n \rangle, \langle \mathit{otherwise}\ B \rangle), \alpha \longrightarrow \square, \gamma & \\[1em]
(\mathit{BPEL}9) & \mathit{Pol} \vdash B, \langle\rangle \longrightarrow \boxdot, \gamma \ \Rightarrow \ \mathit{Pol} \vdash \mathit{scope}\ n : (B, C, F), \alpha \longrightarrow \boxdot, (n : C : \gamma).\alpha & \\
(\mathit{BPEL}10) & \mathit{Pol} \vdash B, \langle\rangle \longrightarrow \boxslash, \gamma \ \wedge \ \mathit{res}(\mathit{Pol}, B, \boxslash) \vdash F, \gamma \longrightarrow \boxminus, \beta \ \Rightarrow & \\
 & \mathit{Pol} \vdash \mathit{scope}\ n : (B, C, F), \alpha \longrightarrow \boxminus, \alpha & \text{where } \boxminus \in \{\boxdot, \boxtimes\} \text{ and } \boxslash \in \{\boxtimes, \overline{\boxtimes}\} \\[1em]
(\mathit{BPEL}11) & \mathit{Pol} \vdash \mathit{compensate}, \langle\rangle \longrightarrow \boxdot, \langle\rangle & \\
(\mathit{BPEL}12) & \mathit{Pol} \vdash C, \beta \longrightarrow \boxdot, \gamma \ \wedge \ \mathit{res}(\mathit{Pol}, C, \boxdot) \vdash \mathit{compensate}, \alpha \longrightarrow \boxminus, \langle\rangle \ \Rightarrow & \\
 & \mathit{Pol} \vdash \mathit{compensate}, (n : C : \beta).\alpha \longrightarrow \boxminus, \langle\rangle & \text{where } \boxminus \in \{\boxdot, \boxtimes\} \\
(\mathit{BPEL}13) & \mathit{Pol} \vdash C, \beta \longrightarrow \boxtimes, \gamma \ \Rightarrow \ \mathit{Pol} \vdash \mathit{compensate}, (n : C : \beta).\alpha \longrightarrow \boxtimes, \langle\rangle & \\
(\mathit{BPEL}14) & \mathit{Pol}_1 \vdash \mathit{compensate}, \alpha_1 \longrightarrow \boxdot, \beta_1 \ \wedge \ \mathit{Pol}_2 \vdash \mathit{compensate}, \alpha_2 \longrightarrow \boxdot, \beta_2 \ \wedge & \\
 & \mathit{res}((\mathit{Pol}_1 \parallel \mathit{Pol}_2), (\alpha_1 \parallel_C \alpha_2), \boxdot) \vdash \mathit{compensate}, \alpha \longrightarrow \boxminus, \beta \ \Rightarrow & \\
 & (\mathit{Pol}_1 \parallel \mathit{Pol}_2) \vdash \mathit{compensate}, ((\alpha_1 \parallel_C \alpha_2).\alpha) \longrightarrow \boxminus, \langle\rangle & \text{where } \boxminus \in \{\boxdot, \boxtimes\} \\
(\mathit{BPEL}15) & \mathit{Pol}_1 \vdash \mathit{compensate}, \alpha_1 \longrightarrow \boxtimes, \beta_1 \ \vee \ \mathit{Pol}_2 \vdash \mathit{compensate}, \alpha_2 \longrightarrow \boxtimes, \beta_2 \ \Rightarrow & \\
 & (\mathit{Pol}_1 \parallel \mathit{Pol}_2) \vdash \mathit{compensate}, ((\alpha_1 \parallel_C \alpha_2).\alpha) \longrightarrow \boxtimes, \langle\rangle & \\[1em]
(\mathit{BPEL}16) & \mathit{Pol}_1 \vdash B_1, \alpha \longrightarrow \square_1, (\gamma)\frown(\alpha) \ \wedge \ \mathit{Pol}_2 \vdash B_2, \alpha \longrightarrow \square_2, (\beta)\frown(\alpha) \ \Rightarrow & \\
 & (\mathit{Pol}_1 \parallel \mathit{Pol}_2) \vdash \mathit{flow}(B_1, B_2), \alpha \longrightarrow (\square_1 \otimes \square_2), ((\gamma \parallel_C \beta).\alpha) & \\[1em]
(\mathit{BPEL}17) & \mathit{Pol} \vdash B, \langle\rangle \longrightarrow \boxdot, \alpha \ \Rightarrow \ \mathit{Pol} \vdash \{\!| B, F |\!\}, \langle\rangle \longrightarrow \boxdot, \langle\rangle & \\
(\mathit{BPEL}18) & \mathit{Pol}_1 \vdash B, \langle\rangle \longrightarrow \boxtimes, \alpha \ \wedge \ \mathit{res}(\mathit{Pol}, B, \boxtimes) \vdash F, \alpha \longrightarrow \boxminus, \beta \ \Rightarrow & \\
 & \mathit{Pol} \vdash \{\!| B, F |\!\}, \langle\rangle \longrightarrow \boxminus, \langle\rangle & \text{where } \boxminus \in \{\boxdot, \boxtimes\}
\end{array}$$

Fig. 12. Labelled Transition Semantics for BPEL

then regardless of what the status of the second activity is going to be, the sequential composition will also be denied execution or force-terminated.

The next pair of rules, $(BPEL7)$–$(BPEL8)$, considers the case of the conditional composition where the final state of the *switch* activity will depend on the status of the selected activity and whether the latter is permitted, denied or force-terminated. The selection of the particular activity is by case and depends on the truth value of its logical guard. Rules $(BPEL9)$ and $(BPEL10)$ deal with scopes. Let $\langle\ \rangle$ the empty compensation context, Rule $(BPEL9)$ states that if the default activity in a scope is permitted to execute and succeed, then the compensation handler corresponding to it is installed in the compensation context ($(n : C : \gamma).\alpha$). In this case, the outcome of the scope is the same as that of the default activity. Rule $(BPEL10)$ states that if the main activity in the scope is denied execution or is force-terminated (by the policy enforcement point or the BPEL engine) then the fault handler activity takes over execution and the outcome of the scope is that of the fault handler's activity. Note that we assume that the fault handler is never force-terminated, and so it is always either permitted or denied execution.

Rules $(BPEL11)$–$(BPEL15)$ deal with the case of compensations. Rule $(BPEL11)$ states that a *compensate* call in an empty compensation context will always succeed regardless of the security policy (therefore its semantics resemble the semantics of *skip*). Rule $(BPEL12)$ states that if the execution of the head of a compensation context is allowed, then the outcome of a compensation call depends on the outcome of the execution of the tail of the context. Rule $(BPEL13)$ states that if the execution of the head of a compensation context is denied by the policy, then so will be the execution of the overall compensation context. The next two rules deal with the case of parallelism in compensation contexts resulting from paralleling in BPEL activities. Rule $(BPEL14)$ states that if two compensation contexts are allowed to run under their respective policies, then the outcome of the overall compensation context will depend on the outcome of its tail. Conversely, rule $(BPEL15)$ states that if one of the two compensation contexts are denied execution, then both acting as the head of a larger compensation context will cause the latter to be denied as well. Both the last two rules use the parallel composition of compensation contexts operator, $||_C$, which is described in the next rule.

Let ˆ be the symbol for the concatenation of compensation context, Rule $(BPEL16)$ deals with the parallel composition of activities using the *flow* activity. There are a couple of interesting notes on this rule. First, the special operation $\otimes$ is used to propagate the outcome (permission, denial or force-termination) of one activity to another in parallel with it.

This operator then determines the outcome of the overall composition. The second point is related to the fact that any new compensation contexts generated by the parallel activities must be treated as being in parallel as well. Therefore, these are composed using a special syntactic operator, $||_C$, to indicate that these must be dealt with in parallel and each under its own security policy.

Finally, we can define now the meaning of a business process, $\{|B, F|\}$, under the control of a policy, *Pol*. This meaning is defined in rules $(BPEL17)$ and $(BPEL18)$. Rule $(BPEL17)$ states that if the main activity B of the business process is permitted by the security policy, then the business process is also permitted by the policy to execute. Rule $(BPEL18)$, on the other hand, states that if activity is denied at any stage, then the fault handler of the business process takes over, under control from the residue of the policy at the point the business process was denied execution. The outcome of the business process will be the same as the fault handler's outcome. Again, there is an assumption here that a fault handler is never force-terminated.

Spatiotemporal Connectives for Security Policy in the Presence of Location Hierarchy

Subhendu Aich[1], Shamik Sural[1], and A.K. Majumdar[2]

[1] School of Information Technology
[2] Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur, India
{subhendu@sit,shamik@sit,akmj@cse}.iitkgp.ernet.in

Abstract. Security of a computer system frequently considers location and time of requesting an access as important factors for granting or denying the request. The security policy specified for such systems should include formalism for representing spatial and temporal facts. Existing security policy specifications already consider useful temporal facts like Monday, Weekends, College hours, etc. We have taken a new approach for representing real world spatial objects in security policy. The proposed representation uses six policy connectives ***at, inside, neighbor, is, crosses, overlapping*** for expressing useful relations existing between practical spatial entities like office, department, roads, etc. The expressiveness of the connectives has been discussed and a formalism for combined spatiotemporal interaction has also been proposed in this paper.

Keywords: Spatiotemporal event, Policy connectives, Location hierarchy, RBAC.

1 Introduction

Access control decision of the currently available security systems depends on context factors of both subject and object involved in a particular access request. The context information includes user location, object location, access time, etc. Some examples of security requirements that the current security systems are supposed to handle are as follows:

- Students are allowed to download bulk data from the Internet only at night.
- Students can access resources only from laboratory computers.
- During weekends, any professor can access resources from his home.

For the above security requirements, we need a security policy which conveniently represents both real world locations and time. Capturing occurrence of event in a particular place and at a particular time, has recently drawn interest of information security researchers. A formalism for combined space and time representation is thus considered to be important and useful. Spatiotemporal interaction has also been found relevant in various disciplines including spatial and temporal databases as well as different GIS services.

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 118–127, 2008.

Several attempts have been made to formally specify time and space [1]. The different approaches by different parties have already raised the inter operability issue. Standard bodies have been formed so that all can work together for a common format [2]. Nonetheless, till now the aspect of representation of space and time interacting with each other has got less attention in existing security policy literature. On the other hand, we believe that occurrence of an event is what is fundamentally relevant to any access control system. Any event normally takes place at precise time and space points. So a formalism for space and time is necessary. Abstract representation of space and time in mathematics (especially in Geometry) is well known. However, a similar abstract representation makes the job of writing spatiotemporal events quite difficult. So what we need is a representation which uses natural language keywords for representing these two interrelated dimensions. At the same time, we should be careful that such a representation does not leave any ambiguity that occurs frequently in the use of natural language. In this paper we present an approach for formalizing space time interaction using spatiotemporal connectives evolved from natural phenomena. In doing so we have modeled spatial objects hierarchically related to each other.

We discuss related work done in this area in the next section. Then we explain the notion of space time interaction in Section 3. In Section 4, we put our space time formalism in place. Section 5 presents the proposed policy connectives in detail. Some examples of requirements in access control have been expressed using our specification in Section 6 and we conclude in Section 7.

2 Related Work

Niezette and Stevenne [1] pointed out that storing and handling of temporal information has been a topic of interest in database technology for many years. The earlier models could not handle the infinite nature of time properly. Without prior knowledge of upper and lower bounds, storing information in a database which repeats, say every month, was difficult or it used to consume a large amount of space. The concept of *generalized database* by Kabanza et al. [3] first proposed linear repeating point for finite representation of infinite temporal data. A *generalized tuple* represents a possibly infinite set of classical tuples. The symbolic representation of periodic time was proposed by Niezette and Stevenne [1]. This representation uses a natural calender for expressing periodic time and was found to be very useful in the context of expressing access requirements. Based on this symbolism, Bertino et al. [4] proposed a temporal authorization system for databases. The same symbolism was subsequently found suitable for expressing temporal constraints for Role Based Access Control model [5].

For formalizing spatial entities, there is an open standard group of body called Open Geospatial Consortium (OGC) [2]. OGC recognizes *features* as the core component for capturing geographic objects of interest [6]. There is another set of documents related to implementation of OGC features which is based on the abstract specification. Such a standard body formalizes the representation of spatial objects. On the other hand, the recent access control models already consider user location for deciding access request. Ray et al. proposed and formalized Role Based Access Control model where access permission is location aware [7]. The model considers an abstract view of the location objects and includes the set based intersection and membership operations only. Another access control model called GEO-RBAC was proposed by Damiani et al. [8] based on the OGC feature specification. Ray and Toahchoodee very recently proposed a protection model called STRBAC [9] for access control in spatiotemporal domain. We have recently proposed a model called STARBAC [10], which extends classical RBAC based on spatiotemporal constraints.

3 Relating Space and Time

Any representation of space and time symbolizes a set of space-time points. We start with an arbitrary representation of space and time ß. Let us consider two expressions $e_1, e_2 \in$ ß representing *Bank ABC during Monday* and *Bank Manager's Office during Banking hours*. In this example, the use of daily space time facts is considered purposefully to emphasize natural viewpoint of our observation. Once again, in natural language, the **common space-time points** of e_1 *and* e_2 basically represent *Manager's Office in Bank ABC during Banking hours on Monday*. The result expression is obtained by considering common space points and common time points, i.e., the concept of commonality has been applied to space and time zones independently. In general, in any relevant composition ô between e_1 *and* e_2 which gives rise to $e_3 \in$ ß, the space and time points interact independently.

One interesting observation in this context is that the space points relate to space points naturally, e.g., **nearest** road of Bank ABC might imply CDE Road. The time points and space points do not relate to each other unless an 'event' occurs. The occurrence of the event is the fact where space and time points converge and thus can be related. E.g., to find out at what time of the day **heavy traffic congestion** occurs on Road CDE results in time points representing 9.00 am - 10.30 am and 5.30 pm - 7.00 pm. Here traffic congestion on road CDE is the event which relates a set of time points with space points.

The observation above leads to certain natural rules which are important for our formalism:

- There is an expression of time where time points relate to time points through use of natural operations. The examples could be *before*, *after*, *days of month*, *first day of year*, etc.
- There is an expression of space where space points relate to space points through use of natural operations. The examples could be *nearest*, *adjacent*, *crosses*, etc.
- There is an expression which starts with time points (space points) on a **defined measurable event** and results in space points (time points). Example is *Which are the crossings that get heavy traffic congestion between 7.00 am - 9.00 am?*

– The semantics of composition in space time expressions may be applied to the spatial and temporal components orthogonally.

4 Space Time Representation

The discussion above gives a new direction for representing space and time. The formalism we present here is orthogonal in space and time. Another important aspect of the representation is listing possible spatial and temporal relations among themselves. So we start by separately stating the symbolism for Time and then for Space. At the same time, we derive a number of natural spatial connectives relating time-time and space-space. Then we frame a number of security policy statements useful in spatiotemporal authorization domain based on the proposed representation.

4.1 Representation of Periodic Interval

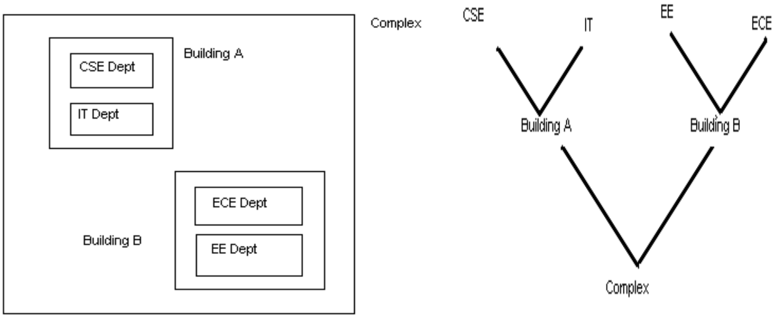
The representation of time is based on the formalism proposed by Niezette and Stevenne [1] using the notion of natural calendars e.g., "every day", "every week", etc. A calendar C is a sub calendar of another calendar D if each interval in D can be covered by some integer number of intervals of C and it is written as: $C \sqsubseteq D$. Bertino et al [4] refined the previous work and proposed symbolic representation for expressing periodic time intervals. The periodic intervals are "Mondays", "the 3rd week of every year", etc.

Periodic Expression: [1] Given calendars, a periodic expression is defined as: $P = \sum_{i=1}^{n} O_i.C_i \triangleright r.C_d$, where $O_1 = all, O_i \in 2^N \cup all, C_i \sqsubseteq C_{i-1}\ for\ i = 2, 3, ..., n, C_d \sqsubseteq C_n\ and\ r \in N$. The symbol $\triangleright$ separates the first part of the periodic expression, identifying the set of starting points of the interval it represents, from the specification of the duration of the interval in terms of calendar C_d. For example, *all.Years + [5,12] .Months* $\triangleright$ *10.Days* represents the set of intervals starting at the same instant as the starting point of the months May and December every year and having a duration of ten days. The scope of P is represented by the bounds *begin* and *end* which is a pair of date expressions of the form $mm/dd/yyyy : hh$ where *end* value can as well be infinity (∞). The periodic time expression looks like $< [begin, end], P >$ or $< I, P >$.

4.2 Representation of Space

We build the formal representation of spatial entities on top of the standard abstract specifications published by OGC [2]. This representation assumes **feature** as the central point for capturing and relating real world spatial entities. Here we describe the core ideas of our formalism.

Logical Location set ($\mathcal{F}$)**.** We treat the logical location elements as instances of feature [6]. Let us consider an academic institute campus map as shown in Figure 4.2. We show only the portion of the campus corresponding to the complex where four academic departments are situated in two separate buildings.



(a) Academic Complex Map

(b) Complex Hierarchy

CSE department, ECE department, Building A and Complex are some examples of logical locations in the campus. There are both spatial and nonspatial characteristics associated with a logical location. But only the spatial attributes of a logical location (the so called geospatial extent of a feature) bear relevance to us here. It is evident that the set of all logical location elements is application dependent and is referred to as $\mathcal{F}$. Any organization is naturally divided into some logical zones.

Location Type set (Ω). The logical location elements in $\mathcal{F}$ can always be attributed to a particular location type. A location type is analogous to a feature type [6]. The feature types express the commonality among elements in $\mathcal{F}$. The set of all defined location types is Ω. Depending on the need of an organization, user defined elements can be added further to Ω. We assume that each logical location element contains an attribute *ft_FeatureType* which is the corresponding location type.

Location in presence of Hierarchy. The types of location we model usually show geometric relation with each other. Before stating the relationship detail we assume that each spatial location is abstracted as a two dimensional object (a reasonable assumption from the point of view of a map). So the objects like roads and buildings in $\mathcal{F}$ are also considered to be two dimensional in nature. The locations in our model maintain hierarchical relationship based on the well defined subset superset operation. Figure 4.2 represents the location hierarchy corresponding to the campus map in Figure 1. When we look down the hierarchy we traverse the path: IT department → Building A → Complex. When we drill down, we move in the other direction. It is perfectly possible that two locations in $\mathcal{F}$ are not hierarchically related at all.

5 Policy Connectives

Each element in $\mathcal{F}$ we consider has got a geospatial extent. We assume that the geospatial extent is specified using feature with geometry [6]. In the proposed representation, the attribute $FT\_FeatureExtent$ of a logical location returns the spatial extent of that feature. The geometry model we use is the Simple Feature Geometry specified in OGC implementation specification document [11]. The base Geometry class has subclasses for Point, Curve, Surface and GeometryCollection. The geometric objects are associated with a Spatial Reference System, which describes the coordinate space in which the geometric object is defined. The interesting part about this implementation specification is that, it has provided methods for testing useful spatial relations existing between geometric objects. The relations are *Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps, Relate, LocateAlong, LocateBetween.* Once again we focus on relationships between two dimensional objects only. These methods behave like boolean predicates and return binary answer when invoked with appropriate input parameters. The semantics of the methods are according to the standard specification [11].

5.1 Definitions

Based on these specified methods we define a set of six policy connectives between the elements of $\mathcal{F}$ and Ω. The proposed connectives relate one subset of locations to another subset of locations. Five out of the six connectives have one strict version and a weak version. The strict version specifies a more restrictive relation than the weak version, whereas the weak version subsumes the result of corresponding strict variation. The actual semantics of the connectives are described in many sorted first order logic (FOL) formula. In all the definitions below, L represents a subset of location set $\mathcal{F}$ and lt stands for an element of location type set Ω.

CONNECTIVE at. This connective searches for a particular location type within a specified boundary. It is a narrowing down search. The strict version at_s returns the set of locations of a particular location type (specified as right operand) contained inside each of the locations specified in the left operand set. The connective is defined as: $L\ at_s\ lt = \{f | f \in \mathcal{F}\ \wedge\ f.ft\_FeatureType = lt\ \wedge\ \forall l \in L\ \mathbf{Contains}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

The weak version at_w returns the set of locations of a particular location type (specified as right operand) contained inside either of the locations specified in the left operand set. The connective is defined as: $L\ at_w\ lt = \{f | f \in \mathcal{F}\ \wedge\ f.ft\_FeatureType = lt\ \wedge\ \exists l \in L\ \mathbf{Contains}(l.ft\_FeatureExtent,\ f.ft\_Feature Extent)\}$.

CONNECTIVE inside. The *inside* connective is semantically opposite to *at* connective and searches by looking up in the location hierarchy. The strict version $inside_s$ returns the set of locations of a particular location type (specified as right operand) containing each of the locations specified in the left operand set i.e., each element returned is a container of all the operand locations. The connective is defined as: $L\ inside_s\ lt = \{f | f \in \mathcal{F}\ \wedge\ f.ft\_FeatureType = lt\ \wedge\ \forall l \in L\ \mathbf{Within}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

The weak version $inside_w$ returns the set of locations of a particular location type (specified as right operand) containing either of the locations specified in the left operand set i.e., each element returned is a container of at least one operand location. The connective is defined as: $L\ inside_w\ lt = \{f | f \in \mathcal{F} \ \wedge\ f.ft\_FeatureType = lt \ \wedge \ \ \exists l \in L\ \mathbf{Within}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

CONNECTIVE neighbor. This connective is for relating neighbor locations which are adjacent to each other but do not overlap. The strict version $neighbor_s$ returns the set of common adjacent locations of all the operand locations. The connective is defined as: $neighbor_s\ L = \{f | f \in \mathcal{F} \ \wedge\ \forall\, l \in L\ \mathbf{Touches}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

The weak version $neighbor_w$ returns each of the adjacent locations corresponding to at least one operand location. The connective is defined as: $neighbor_w\ L = \{f | f \in \mathcal{F} \ \wedge\ \exists\, l \in L\ \mathbf{Touches}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

CONNECTIVE is. This is for location type searching. It returns the subset of location set specified as left operand which are of location type as mentioned in the right operand. The connective is defined as follows L **is** $lt = \{f | f \in \mathcal{F} \ \wedge\ f.ft\_FeatureType \ = \ lt\}$.

CONNECTIVE overlapping. This connective is meant for relating the overlapping locations. Though we are mainly concerned with the hierarchy based relationships, this connective arises as a natural extension of set intersection. The strict version $overlapping_s$ relates the set of locations of a particular type (specified as right operand) spatially overlapping with all of the locations specified in the left operand set i.e., each element returned intersects with all of the operand locations. The connective is defined as: $L\ overlapping_s\ lt = \{f | f \in \mathcal{F} \ \wedge\ f.ft\_FeatureType = lt \ \wedge\ \forall l \in L\ \mathbf{Overlaps}(l.ft\_FeatureExtent, f.ft\_FeatureExtent)\}$.

The weak version $overlapping_w$ relates the set of locations of a particular type (specified as right operand) spatially overlapping with either of the locations specified in the left operand set i.e., each element returned intersects with at least one of the operand locations. The connective is defined as: $L\ overlapping_w\ lt = \{f | f \in \mathcal{F} \ \wedge\ f.ft\_FeatureType = lt \ \wedge\ \exists l \in L\ \mathbf{Overlaps}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

CONNECTIVE crosses. This connective is for relating locations having linear geometry. Location of type Road, Street, River have got these special relation with each other. The strict variation $crosses_s$ relates the set of locations of a particular type (specified as right operand) spatially crossing each of the specified locations mentioned in the left operand set. The connective is defined as: $L\ crosses_s\ lt = \{f | f \in \mathcal{F} \ \wedge\ f.ft\_FeatureType = lt \ \wedge \quad \forall l \in L\ \mathbf{Crosses}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

The weak variation $crosses_w$ relates the set of locations of a particular type (specified as right operand) spatially crossing at least one of the linear locations specified in the left operand set. The connective is defined as: $L\ crosses_w\ lt = \{f | f \in \mathcal{F} \wedge f.ft\_FeatureType = lt \wedge \quad \exists l \in L\ \mathbf{Crosses}(l.ft\_FeatureExtent,\ f.ft\_FeatureExtent)\}$.

5.2 Expressiveness

The above mentioned fundamental policy connectives can be composed meaningfully to unambiguously relate one space with another. The composition is shown to be powerful enough to express useful hierarchical relations existing between real world spatial entities. The composition evaluation is done carefully to avoid any scoping ambiguity. As is evident, the unary connective is high in the precedence relation than the binary connective. Here we provide a few examples of composed policy connectives.

Example 1. Let us assume that there are feature types defined for complex, building, entry point. The complex contains buildings and each building has different entry points. If we want to specify all the entry points in all of the buildings in PQR Complex, we can do so by the following composition.

$$((\text{PQR Complex } at_s \text{ building}) \; at_w \text{ entrypoint})$$

Example 2. Let us assume that the system security officer of the town GHF is interested in finding out all the main roads which meet either of the Highways passing by GHF. He can do so by writing a policy statement like

$$((\text{GHF town } at_s \text{ Highway}) \; crosses_w \text{ road})$$

5.3 Connective Properties

It may be noted that except the *neighbor* connective, every other connective is binary. The connectives are of the form: $2^{\mathcal{F}}$ $Connective$ $\Omega = 2^{\mathcal{F}}$. Starting from a set of locations, it reaches another set spatially related to the previous set. It is observed that *at* and *inside* connectives are semantically opposite. The relation is formally expressed by the following equivalence equation.

$$x \in (l \; inside_w \; lt_x) \Leftrightarrow l \in (x \; at_w \; lt_l), \text{ where } lt_x, lt_l \text{ are location types of } l \text{ and } x \text{ respectively.}$$

The *at* and *inside* connectives are applicable to location hierarchy. Thus the two connectives also show **transitivity** in their application. The complexity of the policy evaluation depends on the underlying mechanism used to solve the connectives. In our case we intend to use the standard GIS implementation. In general, the connectives like *at* and *inside* are bounded by $O(|L|.|\mathcal{F}|)$ steps.

5.4 Combined Space Time Condition

Finally, we define ***spatiotemporal zone*** as a doublet in the form: $< S, T >$ where S is a spatial expression and T is periodic time interval.

Example 3. $< ScholarsRoad \; crosses \; road, 5.00 \, pm - 7.00 \, pm \; on \; weekdays >$
The S and T components in spatiotemporal zone are pure spatial and temporal expressions. These components do not mingle with each other and they take part in any meaningful composition without affecting each other.

6 Framing Security Policy Using Spatiotemporal Connectives

In a typical resource protection environment where both subject and object can be mobile in nature, access control over space and time is a necessity. The authorization domain in such a system spans over the whole spatiotemporal zone. The formalism stated here can help in writing spatiotemporal authorization policies.

fixed space periodic time security policy: Here the space we have considered is a fixed space, whereas the time interval is periodic in nature. The spatiotemporal zone defined in Sub-section 5.4 helps in policy specification.

Example 4. Let us assume one such application where the Access Control Manager in the organization ABC wants to specify a policy for enabling the employee role (say *empABC*) only inside the company building and also during the office hours. The office hour is defined as 8.00am - 5.00pm, Monday to Friday. A spatiotemporal policy statement which specifies such a requirement is of the form shown below

$$\text{enable empABC} < \text{ABCOffice, OfficeHours} >$$
$$\text{where Office Hours} = \text{all.Weeks} + [2..6].\text{Days} + 9.\text{Hours} \triangleright 9.\text{Hours}.$$

inside_at place fixed time security policy: This can be used for specifying event based security policy at a given space time. Based on the occurrence of a certain event at a given place, an authorization restriction is triggered at particular (**at** places) points in homogeneous locations for some amount of time.

Example 5. Let us assume that an unauthorized access (event e) occurs in a lab of ECE department (Figure 1). The institute security policy bars the server access permission in all the departments of the same building (here Building B) for next 8 Hours till the intrusion analysis is resolved. The following policy statement specifies the required authorization.

$$\text{e} \Longrightarrow \text{Restrict Server Access ((ECE department } inside_w \text{ building) } at_w \text{ department) for next 8 Hours.}$$

7 Conclusion

We have shown that the proposed spatial connectives along with the periodic time formalism is capable of expressing various useful spatiotemporal authorization policy statements. The next step would be to use it in building a spatiotemporal security model and its complete analysis. Another point to be noted here is that, during formalization of space, the location objects are flattened into two dimensions. In reality, we need to incorporate spatial entities of complex geometric extent where this assumption may not remain valid. There are useful spatial relations like **above** and **below** for three dimensional objects which could be important for location aware security. We intend to strengthen the model in this respect also in future.

Acknowledgement

This work is partially supported by research grants from the Department of Information Technology and the Department of Science and Technology, Government of India, under Grant Nos. 12(34)/04-IRSD and SR/S3/EECE/082/2007.

References

1. Niezette, M., Stevenne, J.: An Efficient Symbolic Representation of Periodic Time. In: Proc. of the International Conference on Information and Knowledge Management (1992)
2. OpenGIS Consortium, http://www.opengeospatial.org
3. Kabanza, F., Stevenne, J., Wolper, P.: Handling Infinite Temporal Data. In: Proc. of the ACM Symposium on Principles of Database Systems, pp. 392–403 (1990)
4. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: An Access Control Model supporting Periodicity Constraints and Temporal Reasoning. ACM Transactions on Database Systems 23(3), 231–285 (1998)
5. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A Temporal Role-Based Access Control Model. ACM Transactions on Information and System Security 4(3), 191–223 (2001)
6. OpenGIS Consortium. The OpenGIS Abstract Specification. Topic 5: Features, Version: 4. OpenGIS Project Document Number 99-105r2
7. Ray, I., Kumar, M., Yu, L.: LRBAC: A Location-Aware Role-Based Access Control Model. In: Bagchi, A., Atluri, V. (eds.) ICISS 2006. LNCS, vol. 4332, pp. 147–161. Springer, Heidelberg (2006)
8. Damiani, M., Bertino, E., Catania, B., Perlasca, P.: GEO-RBAC: A Spatially Aware RBAC. ACM Transactions on Information and System Security 10(1), Article No. 2 (2007)
9. Ray, I., Toahchoodee, M.: A Spatio-Temporal Role-Based Access Control Model. In: Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pp. 211–226 (2007)
10. Aich, S., Sural, S., Majumdar, A.K.: STARBAC: Spatiotemporal Role Based Access Control. In: Proc. of the Information Security Conference, pp. 1567–1582 (2007)
11. OpenGIS Consortium. OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture. Document Reference Number OGC 06-103r3

BusiROLE: A Model for Integrating Business Roles into Identity Management

Ludwig Fuchs[1] and Anton Preis[2]

[1] Department of Information Systems,
University of Regensburg, 93053 Regensburg, Germany
Ludwig.Fuchs@wiwi.uni-regensburg.de
[2] Institute of Management Accounting and Control
WHU – Otto Beisheim School of Management, Burgplatz 2, 56179 Vallendar, Germany
Anton.Preis@whu.edu

Abstract. The complexity of modern organisations' IT landscapes has grown dramatically over the last decades. Many enterprises initiate role projects in order to reorganise their access structures based on an organisation-wide Identity Management Infrastructure (IdMI). This paper surveys role models and related literature and identifies different role properties. It shows that current role models are not feasible for their usage in IdMIs. By implementing one single type of role they fail to take business requirements and different role perceptions into account. This paper improves the current situation by developing busiROLE, a role model that integrates various types of roles, fulfilling business- as well as IT requirements, and is hence usable in IdMIs.

Keywords: Identity Management, Business Roles, Compliance, IT security.

1 Introduction and Motivation

Large companies have to manage complex organisational structures and a large number of identities within their IT systems. As a result of incorrect account management users accumulate a number of excessive rights over time, violating the principle of the least privilege [1]. This situation results in a so called identity chaos. Implementation projects like [2] and studies [3] show that major security problems arise because of employees gaining unauthorized access to resources. National and international regulations like Basel II [4], the Sarbanes-Oxley Act [5], and the EU Directive 95/46 [6] together with internal guidelines and policies force enterprises to audit the actions within their systems. Roles are seen as means to meet compliance demands in general. Yet, implementing a technical IdMI as presented in [7] is only the starting point for getting compliant. IdM is not able to take business needs into consideration on a purely technical level. Organisational IdM integrates business requirements into global user management processes. Its understanding of roles is shifted from a rights-based towards a task- and organisation-oriented role concept [8]. Nevertheless, companies and IdM vendors mainly implement a basic role model [9] which defines one single type of role only. The main problem is that various types of roles, e.g. Business

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 128–138, 2008.

Roles, Organisational Roles, Basic Roles, or IT Roles exist within the company without any model that defines and standardises the relations between them. However, as IdM has the goal to essentially connect the technical IT layer with the business perspective, it needs to be able to integrate these different kinds of roles. Bertino et al. [10] likewise mention that Enterprise Security Management tools like IdM solutions don't conform to basic Role-Based Access Control (RBAC) even though they are generally considered to be among the most important RBAC applications. The goal of this paper is to improve that situation by introducing busiROLE, a role model which integrates the different types of roles needed in IdMIs. BusiROLE is also currently used as the underlying formal model during the process of role development and the whole lifecycle of a role system as presented in [11].

This paper is structured as follows. In section 2, related work is presented. A survey of existing role models and properties in section 3 gives an overview and a classification of well-known properties of classic role models. Section 4 subsequently introduces busiROLE explaining the different components, showing their peculiarities and relationships. Finally, conclusions and future work are given in section 5.

2 Related Work

2.1 In-House Identity Management

Over the last few years in-house IdM, i.e. Identity Management within the IT infrastructure of companies, has established itself as a core component of Enterprise Security Management. It deals with the storage, administration, and usage of digital identities during their lifecycle. The aforementioned identity chaos needs to be faced by implementing a centralised IdMI as shown in [7]. Its main building blocks are a Directory Service, User Management, Access Management, and an Auditing Module. Directory Services provide synchronised identity information that is facilitated by the other components. User Management e.g. deals with the provisioning of users, granting and revoking access to resources. When users logon to certain applications, Access Management controls the access to the requested resource while users' as well as administrators' activities are logged within the Auditing Module. IdM duties cover rather simple tasks like automatic allocation and revocation of user resources. However, they also include sophisticated tasks like role management.

2.2 RBAC and Role Types

Role-Based Access Control is a widely used access control paradigm. In its original sense users are assigned to roles and roles are associated with permissions that determine what operations a user can perform on information objects acting as a role member. Besides a more secure and efficient user- and resource management, manual administration efforts are minimised and compliance issues addressed by the usage of roles [12]. Numerous role models have evolved as a result of special industry needs. Additionally, the difference between IT- and business- related roles has been discussed intensively [1]. Roles on the IT layer are essentially bundles of permissions within an application. Business related roles are defined on work patterns, tasks, and the position of employees within the organisation. Both concepts can be connected by

Table 1. Role characteristics according to application layer

Criterion	*Business layer*	*IT layer*
Role concept	Organisation-, task-, competence-oriented	Rights-based
Application area	Business processes, workflows, task bundles	Local application, IT system
Responsibilities	Business manager, process owner	IT administrator

defining the permissions that are needed to perform the various tasks. Table 1 gives a short overview over their main characteristics according to their application level.

Adjacent research areas, e.g. company-wide authorisation models and role engineering approaches work with a business-related perception. The Stanford Model [13] for instance integrates *tasks* and *functions* of employees in its architecture. Even though this model attempts to manage the relationships between components in a structured way, its complexity makes the adoption in an IdMI hardly manageable. Wortmann [14] introduces the concept of *person*, *process-role*, and *tasks* in his approach. Still, coming from a process-oriented background, he focuses on the operational structuring and omits organisational structures. Yet, this integration is of major importance for organisational IdM. Some role engineering approaches like [15] or [16] also work with a business-related definition of the term *role*. Epstein [15] introduces entities like *job* or *workpattern* but does not relate them to business needs.

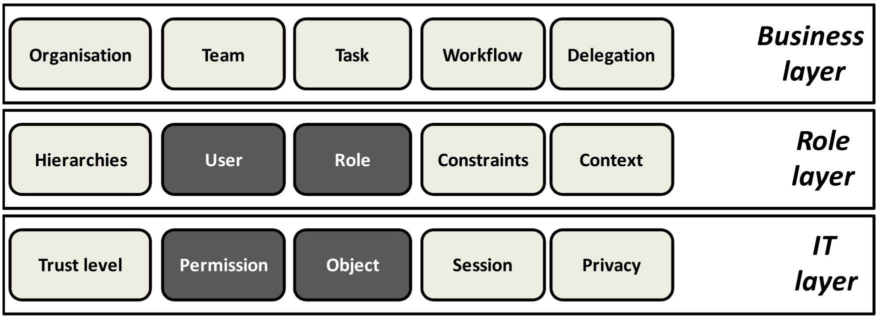


Fig. 1. Role properties and their classification. The IT layer represents the system resources. The Role layer acts as intermediary between IT- and organisational aspects. Role layer properties can be closer to business or IT aspects or even appear on all layers. The Business layer represent both the static (organisational structure) and dynamic aspects (operational structure) of organisations and their interdependencies known from organisational theory [17].

3 Role Properties – Overview, Classification, and Survey

In order to define a generic role model in terms of organisational IdM we start by analysing role properties. Each role model implements a subset of those properties, interpreting the term *role* in its own notion. We use a classification splitting the organisation into a Business-, a Role-, and an IT layer (Fig. 1). Note that properties are classified according to their usage in the according role models. Privacy or Trust, e.g., can be regarded as business- related but are used in a technical manner in the surveyed models. Hence they are located at the IT layer. In general this framework firstly relates role properties to the corresponding layer and secondly differentiates between core- and extensional properties. This way we are able show whether a role model is rather resource- or business- oriented. Analysing 17 major role models we were able to identify 15 different role properties. *Role*, *User*, *Permission*, and *Object* are the core components of every surveyed model. Additional properties surround these central components leading to a functional extension of a role model. Hence they are classified as extensional properties. In the following we are going to shortly present the properties known from the RBAC literature in tables 2 – 4.

Table 2. IT layer Properties

Property	Description
Object (OBJ)	Represents a system resource. The collectivity of all objects represents the set of all resources of an IT system operations can be performed on.
Permission (PRMS)	Represents the right to perform an operation on an OBJ. We represent a permission as a pair *(am, o)* where *am* is an access mode, identifying a particular operation that can be performed on the object *o*.
Session (SESSION)	Sessions are necessary to enable and document user and role activities. Once a role is activated, a session has to be started.
Trust levels (TRUST)	Trust levels help to differentiate between security levels. Objects as well as roles can have different trust levels. Trust levels of objects must be determined whereas role trust levels can e.g. be earned by trustworthy behaviour. Trust levels can be modelled as attributes, for instance.
Privacy (PRIV)	Privacy refers to the security of personal data. Models with a privacy property refer to the protection of personal related data. Similar to TRUST, modelling of PRIV can be done using attributes.

Table 3. Role layer Properties

Property	Description
Role (ROLE)	From an IT-related point of view, a role can be a bundle of permissions to operate on certain IT- or systemic objects. From an organisational perspective, however, a role is a link from an organisational property to an employee who for example has to fulfill a certain task within a team or a context.
User (USER)	A person who is assigned to a certain role. From a more systemic point of view a user needn't necessarily be human and can even be part of an IT process that is assigned to a role.
Hierarchies (HIER)	Among roles, there can be hierarchical relations. Permissions can be inherited from one role by another. Additionally work can be delegated using a role hierarchy.
Constraints (CONSTR)	Refer to the relations among different role properties. They can appear in every layer. With them, limitations and rules are formed. Other properties, e.g. contexts, can be modeled by constraints.
Context (CTXT)	Represents the circumstances in which roles are activated. An example could be an emergency case where the activation of a number of special roles with clearly defined permissions is necessary.

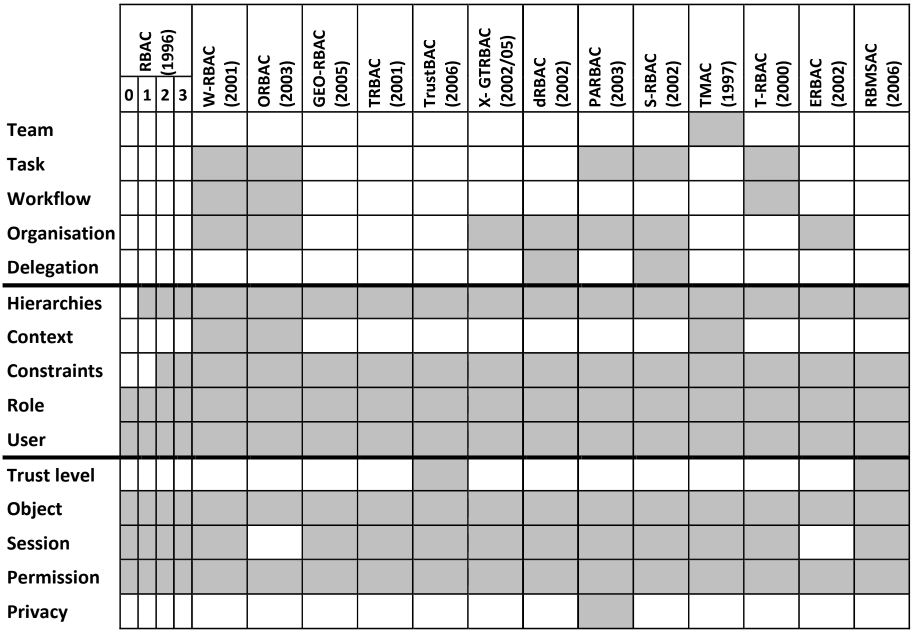
Model Survey and Overview

We are now going to present an abstract of our role model survey. Due to space limitations, we are focusing on the discovered role properties and their usage in existing role models. For the same reason we are additionally not referencing every single model separately in the references section. Many of them can be found in [1]. Throughout the survey, every considered role model has been visualised using thethree-layer classification introduced at the beginning of this section. Figure 2 sums up the classification results. The tableau shows for instance that TrustBAC [19] implements *Role Hierarchies*, *Constraints*, *Sessions*, and extends basic RBAC functionality using system-specific *Trust Levels*. Moreover it points out that none of the business properties are realised in TrustBAC. We are aware that this tableau only gives a qualitative impression as there is no standardised definition used among all models for the single role properties. Nevertheless it points out the core properties that are implemented by all role models. One can also see that most role models are

Table 4. Business layer Properties

Property	Description
Organisation (ORG)	An organisation can be divided into organisational- and operational structure. The latter includes dynamic aspects like TASK, WFL, and partially even TEAM structures. Organisational structures represent the different hierarchy types and their entities within the organisation.
Task (TASK)	Represents a certain job or duty that has to be fulfilled with regards to a specified outcome. A task can consist of several partial tasks or subtasks. Tasks also can be pooled and then form task groups.
Workflow (WFL)	A subtask consists of workflow units. Ultimately, a workflow has to result in a certain outcome, which makes it similar to tasks. Unlike tasks, the focus lies on the sequence of steps to be performed.
Team (TEAM)	A very narrow definition of this property could be a user pool. Teams are task- and goal-oriented [18]. The component can be seen as in-between of organisational and operational structures.
Delegation (DELEG)	The term *delegation* has a two-sided meaning: First, it can be total when e.g. whole roles are delegated from one user to another Second, the delegation can also be partial and be only valid for single tasks. Delegation is closely connected to the role hierarchies from table 3.

IT-oriented. Even though some of them are implementing business properties, e.g. ORBAC [20], none of the models is really business-focused and therefore feasible for role-based IdM deployments. The most powerful model concerning the representation of business properties is the SRBAC model, [21] as it is capable of modelling organisational structure and functional units. It has, however, a limited definition of hierarchies. The central outcome of our analysis is that the models each define only one single type of roles. The basic RBAC family [9], even though it is used in most IdM deployments, can also not meet the requirement of multiple role types. This



	RBAC (1996)				W-RBAC (2001)	ORBAC (2003)	GEO-RBAC (2005)	TRBAC (2001)	TrustBAC (2006)	X- GTRBAC (2002/05)	dRBAC (2002)	PARBAC (2003)	S-RBAC (2002)	TMAC (1997)	T-RBAC (2000)	ERBAC (2002)	RBMSAC (2006)
	0	1	2	3													
Team														▨			
Task					▨	▨						▨	▨		▨		
Workflow					▨	▨									▨		
Organisation					▨	▨				▨	▨	▨	▨			▨	
Delegation											▨		▨				
Hierarchies		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Context					▨	▨								▨			
Constraints			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Role	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
User	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Trust level									▨								▨
Object	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Session	▨	▨	▨	▨	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨		▨
Permission	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Privacy												▨					

Fig. 2. Survey of existing role models. The first vertical column lists the role properties while the horizontal axis represents the various role models. Grey colouring indicates that an according property is realised in a certain role model. Unavailable functionalities remain white.

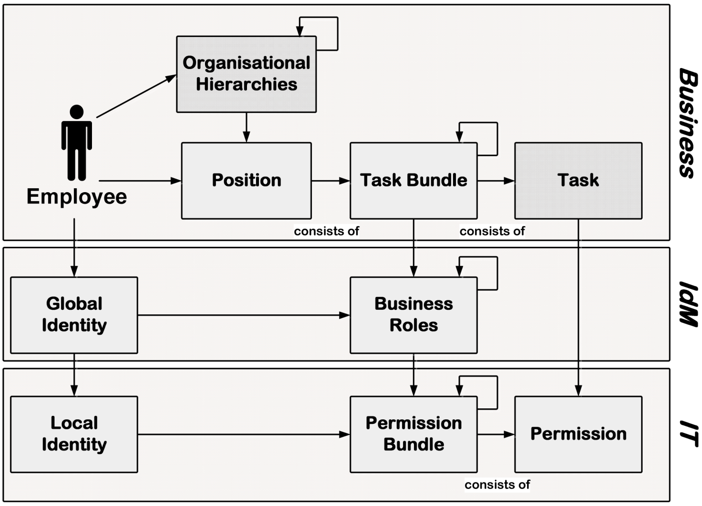


Fig. 3. BusiROLE. The business properties form the organisational basis. Their relationships represent the hierarchical dependencies and an increasing granularity level, from Organisational Hierarchies and Positions towards Task Bundles and singular Tasks.

result complies with and underlines Bertino et al.'s finding [10] that the RBAC family is not feasible for usage in Enterprise Security Management Infrastructures.

4 BusiROLE – Integrating Business Role Types into IdM

In the following we define the properties and the types of roles needed for successful role deployment in Identity Management solutions. We argue that the integration of more than one role type is necessary to take business structures and requirements into account. In busiROLE every employee has a number of different roles that stem from his position in various Organisational Hierarchies and his assigned Task Bundles. Figure 3 shows the busiROLE core entities and role types derived: On the Business layer we integrate different types of *Organisational Hierarchies* (Basic Roles), *Positions* (Organisational Roles), *Task Bundles* (Functional Roles), and *Tasks*. The *Business Roles* entity is the core component of our model. It represents the collectivity of an employee's roles. Note that we also could have modelled each role type separately and connected it with the corresponding business entity. However, for clarity reasons the Business Roles entity bundles all the different role types seen in figure 4. We furthermore introduce an *Employee* entity and an according *Global Identity* that links to all application-specific user accounts (*Local Identity*). On the IT layer we use *Permission Bundles* which can be expressed e.g. using local IT Roles. This feature is needed to connect local systems with different permission handling mechanisms to a global IdMI. However, we are not going into detail about the IT layer elements as we adopt the well-known resource-oriented RBAC-approach [9] on this layer.

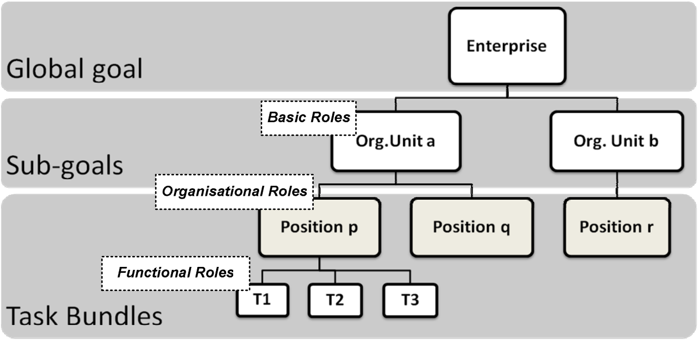


Fig. 4. Organisational goal, Sub-goals, and Task Bundles. As long as a goal is too complex for the assignment to an employee we speak of goals and sub-goals. Sub-goals are split into Task Bundles that are assigned to certain positions.

4.1 Business Layer Properties

According to organisational theory a firm's global goal is split up into sub-goals and assigned to single organisational units (see figure 4). Employees' responsibilities are defined by what they do, who they report to, and who reports to them. These definitions are assigned to positions rather than to specific individuals. The positions are in turn assigned to predefined work packages. Scientific publications in the business administration area and in the area of organisational behaviour contain a profound theoretical insight into relevant facts and relations within an organisation [17], [22]. Our survey in section 3 has shown that up to now no suitable role model for IdMIs exists, mostly because of missing a differentiation between role types and hence business layer properties. We argue that organisational structure, rather than operational structure, is the main pillar of a role model in IdMIs. Operational structures, i.e. process-, workflow-, and single task definitions are not feasible within IdMIs. On the one hand it is not possible to keep a complete process- and workflow database up-to-date within an organisation. On the other hand existing IdMIs already are closely related to the line organisation making it easily extensible.

Employee

BusiROLE needs to be capable of assigning existing Positions and Task Bundles to the according persons based on different hierarchy types within the enterprise. We hence extend and split the *User* concept known from the original classification in figure 1: The business property *Employee* is introduced as the counterpart of a *Global Identity* representing the core user account within the IdMI on the Role layer. *Local Identities* are the user accounts of employees on the IT layer. This structuring is already well known from the implementation in most IdM solutions.

Organisational Hierarchies (Basic Roles)

As mentioned beforehand, IdM solutions are already closely aligned to the organisational structure of an enterprise. Organisational Hierarchies can be used to represent the Basic Roles of employees, i.e. permissions that every employee in a certain organisational unit is granted. The *Organisation* property from our original classification, however, has to be extended in order to be able to represent any kind of hierarchical structure. Besides the line organisation IdM solutions have to be able to integrate e.g. a financial- or a reporting hierarchy. A team can also be regarded as a (temporary) organisational unit. We hence omit teams as a separate busiROLE entity. Two employees could e.g. have the same position in the line organisation and consecutively the same Functional Roles derived from that position, however, one of them might have a special Task Bundle related to a team where he is member of. This is represented using a different Organisational Hierarchy type which contains existing team structures and related positions as well as Task Bundles.

Position (Organisational Role)
Positions are needed to represent functional entities within organisational hierarchy elements. They are regarded as Organisational Roles and abstract descriptions of a collection of Task Bundles assigned to certain employees. An example would be a *Windows Developer* within the *Development* department. SRBAC [21], e.g., is already working with so called functional units which are able to represent IdM requirements regarding job positions. Note that there is a relationship between the Organisational Roles and the Basic Roles.

Task Bundle (Functional Role) and Task
Taking a closer look at the *Task* property from the classification in section 3 one can see that existing definitions are not able to model Task Bundles, i.e. hierarchical structures within *n* available tasks. Task Bundles are essentially the Functional Roles of employees. Task Bundles are defined by business representatives according to the sub-goal of an organisational unit and the qualification and workload of an employee. Note that they are only assigned to a position if their complexity allows treatment by a single employee. If no Task Bundles are defined for a certain Position, the Position itself is representing the complete Task Bundle of an employee (see figure 4). Note that Task Bundles also might not necessarily be connected with a Position but directly related to an Organisational Hierarchy element. For example, a manager might assign a special duty to one employee independent from his position within the organisation. Another conceivable scenario could be delegated Functional Roles.

4.2 IdM Layer Properties

After having presented the required business properties we are going to examine the IdM layer properties of busiROLE. Note that the definition of the Role layer from our survey differs to the understanding of our IdM Layer: Many of the role models define the Role layer as the Access Control layer on top of the permissions within an application. In our context, the IdM layer is comprised of the Business Roles and their properties managed within the organisation-wide IdMI. Local IT Roles or permission bundles as they are used in the existing models are a part of the IT layer in our approach.

Global Identity
As mentioned beforehand we introduce a global identifier for each employee. Every single application-specific user account is mapped to exactly one global ID in order to be able to activate and deactivate it automatically via resource provisioning processes of the IdM. This feature is well known from available IdM solutions.

Business Roles

We define the *Business Roles* entity as the IdM representation of an employee's Basic-, Organisational-, and Functional Roles. Business Roles essentially connect the task-oriented business view with the resource-oriented IT layer, integrating the different types of roles needed within an IdMI. Additionally, they are able to include technical measures like Trust or Privacy as we know them from our original classification. For usability and management reasons we argue that the granularity level for a direct connection of Business Roles with Tasks is too high as a single Task does not represent an independent role within the enterprise. Hence Business Roles only represent Task Bundles, Positions, and Organisational Hierarchy elements in form of the different role types. BusiROLE directly relates Task Bundles and Business Roles, nevertheless if a company has not defined fine grained Functional Roles (i.e. Task Bundles), Positions are essentially representing the Task Bundle of an employee (see figure 4). However, note that such a situation limits the usability and the flexibility of busiROLE: Delegation could only be conducted on position (i.e. Organisational Role) level and additionally the different permutations of Task Bundles assigned to a Position could not be modelled accurately. Redundancy issues within the role definitions would complicate the role management.

4.3 Global and Extensional Properties

In the following, we are going to introduce two global properties and four extensional, hence not mandatory properties of busiROLE. They are not modelled as entities but as attributes of core entities.

Constraints and Context

Constraints and context are global properties that influence all entities and their relationships. Using the definition given in section 3, constraints can be viewed as conditions imposed on the relationships, assignments, and entities. An Employee acting in the Organisational Role of a financial manager may not be allowed to act as a financial auditor at the same time (separation of duty). We are expressing them in terms of system-, entity-, or relationship policies. Using a context, we can model certain exceptional circumstances in which a particular role can be activated.

Workflow, Delegation, Trust, and Privacy

Those four properties from figure 1 are handled as extensional properties. Delegation functionality can be implemented as an attribute of a Position. Organisational policy defines which employees can be appointed as representative of another employee under exceptional circumstances. Workflows known from various role models like [23] and [24] are not originally integrated within IdMIs. Within bigger companies it is impossible to maintain a workflow base which represents all existing workflows in combination with the needed permissions at a certain point of time. As aforementioned this is an aspect where our model differs from existing operational-based approaches (like [14]).

5 Conclusions and Future Work

This paper has shown that widely used basic RBAC models are not feasible for their usage in modern IdM because each only implements one single type of role. On basis of a short survey we hence presented busiROLE, a role model able to integrate role types needed in organisational IdMIs, namely Basic Roles, Organisational Roles, Functional Roles, and IT Roles. Its biggest advantage compared to existing models is the usability in organisation-wide and application-independent IdMIs. On the basis of busiROLE, modern IdM vendors as well as big enterprises operating an IdMI are able to closely connect business objectives and technical user management. BusiROLE provides the capability to represent business structures within the company-wide IdMI. Hence it fosters the business-oriented development, management, and maintenance of roles. It is furthermore independent from local Access Control Mechanisms, making it easy to integrate several different target systems. It is currently used as the core model of our hybrid role development approach. Future work includes the introduction of customisable position types in order to provide reusable single position patterns. We furthermore are going to investigate the hierarchical relations among the different role types. Besides such theoretical extensions and improvements busiROLE is going to be tested within different Identity Management Infrastructures. This way the advantages of different role types within one IdMI can be made visible.

Acknowledgments. The work reported in this paper will be continued within the SPIKE project which is funded by the European Union under the Seventh Framework Program (Contract No. 217098).

References

[1] Ferraiolo, D.F., Kuhn, R.D., Chandramouli, R.: Role-Based Access Control. Artech House, Boston (2007)

[2] Larsson, E.A.: A case study: implementing novell identity management at Drew University. In: Proceedings of the 33rd annual ACM SIGUCCS conference on User services, Monterey, CA, USA (2005), http://doi.acm.org/10.1145/1099435.1099472

[3] Dhillon, G.: Violation of Safeguards by Trusted Personnel and Understanding Related Information Security Concerns. Computers & Security 20(2), 165–172 (2001)

[4] Bank for International Settlements BIS: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version (2006), http://www.bis.org/publ/bcbs128.pdf

[5] Sarbanes, P.S., Oxley, M.: Sarbanes-Oxley Act of 2002, also known as the Public Company Accounting Reform and Investor Protection Act of 2002 (2002), http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf

[6] European Union: Directive 95/46/EC of the European Parliament and of the Council. Official Journal of the European Communities L (28-31) (1995), http://ec.europa.eu/justice_home/fsj/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf

[7] Fuchs, L., Pernul, G.: Supporting Compliant and Secure User Handling – a Structured Approach for In-house Identity Management. In: Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES 2007), Vienna, Austria (2007), http://dx.doi.org/10.1109/ARES.2007.145

[8] Walther, I., Gilleßen, S., Gebhard, M.: Ein Bezugsrahmen für Rollen in Unternehmungen. Teil 2: Klassifizierung von Rollen und Situationen. Working Paper 1/2004, University of Erlangen-Nürnberg, Department of Wirtschaftsinformatik I (2004), http://www.forsip.de/download.php?file=/publikationen/siprum/iw-sg_arbeitsbericht_2.pdf

[9] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE Computer 29(2), 38–47 (1996)

[10] Li, N., Byun, J., Bertino, E.: A Critique of the ANSI Standard on Role-Based Access Control. IEEE Security & Privacy 5(6), 41–49 (2007)

[11] Fuchs, L., Pernul, G.: proROLE: A Process-oriented Lifecycle Model for Role Systems. In: Proceedings of the 16th European Conference on Information Systems (ECIS), Galway, Ireland (2008)

[12] Gallaher, M. P., O'Connor, A. C., Kropp, B.: The economic impact of role-based access control. Planning report 02-1, National Institute of Standards and Technology, Gaithersburg, MD (2002), http://www.nist.gov/director/prog-ofc/report02-1.pdf

[13] McRae, R.: The Stanford Model for Access Control Administration, Stanford University (unpublished) (2002)

[14] Wortmann, F.: Vorgehensmodelle für die rollenbasierte Autorisierung in heterogenen Systemlandschaften. Wirtschaftsinformatik 49(6), 439–447 (2007)

[15] Epstein, P., Sandhu, R.: Engineering of Role/Permission Assignments. In: Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001), New Orleans, LA, USA (2001), http://doi.ieeecomputersociety.org/10.1109/ACSAC.2001.991529

[16] Roeckle, H., Schimpf, G., Weidinger, R.: Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In: Proceedings of the fifth ACM workshop on Role-based access control, Berlin, Germany (2000), http://doi.acm.org/10.1145/344287.344308

[17] Mintzberg, H.: Structuring of Organizations. Prentice Hall, Englewood Cliffs (1979)

[18] Katzenbach, J.R., Smith, D.K.: The Wisdom of Teams: Creating the High-Performance Organization. Harvard Business School Press, Boston (1993)

[19] Chakraborty, S., Ray, I.: TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In: Proceedings of the eleventh ACM symposium on Access control models and technologies, Lake Tahoe, CA, USA (2006), http://doi.acm.org/10.1145/1133058.1133067

[20] El Kalam, A.A., Benferhat, S., Miege, A., El Baida, R., Cuppens, F., Saurel, C., Balbiani, P., Deswarte, Y., Trouessin, G.: Organization based access control. In: Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, June 2003, pp. 120–131 (2003), http://doi.ieeecomputersociety.org/10.1109/POLICY.2003.1206966

[21] Seufert, S.E.: Der Entwurf strukturierter rollenbasierter Zugriffskontrollmodelle. Informatik – Forschung und Entwicklung 17(1), 1–11 (2002)

[22] Daft, R.: Organization Theory and Design, 2nd edn. West, St. Paul, Minn. (1986)

[23] Wainer, J., Barthelmess, P., Kumar, A.: W-RBAC - A Workflow Security Model Incorporating Controlled Overriding of Constraints. International Journal of Cooperative Information Systems 12(4), 455–485 (2003)

[24] Oh, S., Park, S.: Task-Role Based Access Control (T-RBAC): An Improved Access Control Model for Enterprise Environment. In: Ibrahim, M., Küng, J., Revell, N. (eds.) DEXA 2000. LNCS, vol. 1873, Springer, Heidelberg (2000), http://dx.doi.org/10.1016/S0306-4379-02-00029-7

The Problem of False Alarms: Evaluation with Snort and DARPA 1999 Dataset

Gina C. Tjhai[1], Maria Papadaki[1], Steven M. Furnell[1,2], and Nathan L. Clarke[1,2]

[1] Centre for Information Security & Network Research, University of Plymouth,
Plymouth, United Kingdom
info@cisnr.org

[2] School of Computer and Information Science, Edith Cowan University,
Perth, Western Australia

Abstract. It is a common issue that an Intrusion Detection System (IDS) might generate thousand of alerts per day. The problem has got worse by the fact that IT infrastructure have become larger and more complicated, the number of generated alarms that need to be reviewed can escalate rapidly, making the task very difficult to manage. Moreover, a significant problem facing current IDS technology now is the high level of false alarms. The main purpose of this paper is to investigate the extent of false alarms problem in Snort, using the 1999 DARPA IDS evaluation dataset. A thorough investigation has been carried out to assess the accuracy of alerts generated by Snort IDS. Significantly, this experiment has revealed an unexpected result; with 69% of total generated alerts are considered to be false alarms.

Keywords: Intrusion Detection System, False positive, True positive, DARPA dataset, Snort.

1 Introduction

The issue of false positives has become the major limiting factor for the performance of an IDS [5]. The generation of erroneous alerts is the Archilles' heel of IDS technology, which could render the IDS inefficient in detecting attacks. It is also estimated that a traditional IDS could possibly trigger 99% of fake alarms from total alerts generated [10]. Recognising the real alarms from the large volume of false alarms can be a complicated and time-consuming task. Therefore, prior to addressing the issue of false alarm, a quantitative evaluation is required to assess the extent of the false positive problem faced by current IDS.

A number of research or efforts have been conducted to evaluate the performance of IDS in terms of its detection rate and false positive rate. One of the most well-known and determined IDS assessments to date was undertaken by Defense Advanced Research Projects Agency (DARPA) IDS evaluation [12]. This quantitative evaluation was performed by building a small network (test bed), which aimed to generate live background traffic similar to that on a government

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 139–150, 2008.

site connected to the Internet. The generated data set, which included a number of injected attacks at well defined points, were presented as tcpdump data, Basic Security Model (BSM), Windows NT audit data, process and file system information. The data were then used to evaluate the detection performance of signature-based as well as anomaly-based IDSs [14].

Although this data set appears to be one of the most preferred evaluation data sets used in IDS research and had addressed some of the concerns raised in the IDS research community, it received in-depth criticisms on how this data had been collected. The degree to which the stimulated background traffic is representative of real traffic is questionable, especially when it deals with the reservation about the value of the assessment made to explore the problem of the false alarm rate in real network traffic [16]. Significantly, Mahoney and Chan [15] also critically discuss how this data can be further used to evaluate the performance of network anomaly detector. Although the DARPA IDS evaluation dataset can help to evaluate the detection (true positive) performance on a network, it is doubtful whether it can be used to evaluate false positive performance. In fact, the time span between the dataset creation and its application to the current research has resulted in another reservation about the degree to which the data is representative of modern traffic. However, despite all of these criticisms, the dataset still remains of interest and appears to be the largest publicly available benchmark for IDS researchers [16]. Moreover, it is also significant that an assessment of the DARPA dataset is carried out to further investigate the potential false alarms generated from this synthetic network traffic. It is expected that the result of this analysis could describe or provide a general picture of the false alert issue faced by the existing IDSs.

The main objective of the experiment described in this paper is to explore the issue of false alarm generation on the synthesized 1999 DARPA evaluation dataset. An investigation is also conducted to critically scrutinize the impact of false alarms on the IDS detection rate. Section 2 presents a number of related studies carried out to evaluate the performance of IDS. Section 3 discusses the methodology of the experiment. The findings are presented in section 4 and lastly, followed by conclusions in section 5.

2 Related Works

As for IDS performance, a study has also been conducted to further assess the effectiveness of Snort's detection against 1998 DARPA dataset evaluation [8]. Snort is an open source and signature-based IDS [9]. It is a lightweight IDS which can be easily deployed and configured by system administrators who need to implement a specific security solution in a short amount of time [17]. In other words, Snort is a flexible programming tool which enables the users to write their own detection rules rather than a static IDS tool. The evaluation was performed to appraise the usefulness of DARPA as IDS evaluation dataset and the effectiveness of the Snort ruleset against the dataset. Surprisingly, the result showed that Snort's detection performance was very low and the system produced an unacceptably high rate of false positives, which rose above the 50% ROC's guess line rate. Unfortunately, no further explanation was given to describe the nature of false alarms.

Interestingly, a paper by Kayacik and Zincir-Heywood [11] discussed the benefit of implementing intrusion detection systems working together with a firewall. The paper had demonstrated a benchmark evaluation of three security management tools (Snort, Pakemon and Cisco IOS firewall). Significantly, the result showed that none of the tools could detect all the attacks. In fact, Snort IDS was found to have produced 99% of false alarm rate, the highest rate compared to the other IDS (Pakemon). The result had also revealed that Cisco IOS had performed well and raised only 68% of false alarm rate. This has suggested the implementation of a firewall-based detection, which in turn decreases the attack traffic being passed to the IDSs.

Apart from the two studies above, which focused upon Snort performance, there are a large number of studies that have used the 1998 and 1999 DARPA dataset to evaluate the performance of IDSs. One of those studies is that of Lippmann et al [13], which managed to demonstrate the need for developing techniques to find new attacks instead of extending existing rule-based approach. The result of the evaluation demonstrated that current research systems can reliably detect many existing attacks with low false alarm rate as long as examples of these attacks are available for training. In actual fact, the research systems missed many dangerous new attacks when the attack mechanisms differ from the old attacks. Interestingly, a similar paper had also been written by Lippmann et al [14], focusing upon the performance of different IDS types, such as host-based, anomaly-based and forensic-based in detecting novel and stealthy attacks. The result of this analysis had proposed a number of practical approaches applied to improve the performance of the existing systems.

Alharby and Imai [2] had also utilised 1999 DARPA dataset to evaluate the performance of their proposed alarm reduction system. In order to obtain the normal alarm model, alarm sequence is collected by processing the alerts generated by Snort from the first and third weeks (free-attacks traffic) of DARPA 1999 dataset. From these alarm sequences, the sequential patterns are then extracted to filter and reduce the false alarms. The same dataset (using the first and third weeks of the 1999 DARPA dataset) had also been applied by Bolzoni and Etalle [7] to train and evaluate the performance of the proposed false positive reduction system. Similarly, Alshammari et al [3] had also used such data to experiment their neural network based alarm reduction system with the different background knowledge set. The final result has proved that the proposed technique has significantly reduced the number of false alarms without requiring much background knowledge sets.

Unlike other papers discussed above, our experiment focuses specifically upon the issue of false alarms, rather than the performance of IDS (true alarms) in general. In this study, we propose to investigate in a more detailed manner some of the shortcomings that caused the generation of false alarms.

3 Experiment Description

Given that the 1999 DARPA dataset is deemed to be the largest publicly available benchmark, our experiment was designed to utilize such data as the source of our investigation. The experiment was run under Linux Fedora 7, and Snort version 2.6 was chosen as the main detector. The reason for utilising Snort was due to its openness and public availability. The Snort ruleset deployed in this evaluation is VRT Certified Rules for Snort v2.6 registered user release (released on 14 May 2007). In order to facilitate the analysis of IDS alerts, a front-end tool Basic Analysis and Security Engine (BASE) was utilized as the intrusion analyst console [6].

The primary data source of this evaluation was collected from DARPA evaluation dataset 1999. Without training the Snort IDS with the three weeks training data provided for DARPA off-line evaluation beforehand, two weeks testing data (fourth and fifth week of test data) were downloaded and tested. Snort ran in its default configuration, with all signatures enabled.

The first stage of the experiment was to run Snort in NIDS mode against the DARPA dataset. The manual validation and analysis of alerts produced by Snort were undertaken by matching against the Detection and Identification Scoring Truth. The Detection Scoring Truth is comprised of a list of all attack instances in the 1999 test data, while Identification Scoring Truth consists of alert entries of all attack instances in the test data [12]. A match is identified as same source or destination IP address, port numbers and their protocol type. In this case, timestamp does not really help identifying the true alerts since the attacks were labeled by the time the malicious activities set off while Snort spotted them when malevolent packets occurred. This might render the system missing numerous matches. Hence, by recognizing the matches for those attack instances, the number of false positives alarms will then be identified.

Once the alerts were manually verified and the false positives were isolated, the results were presented in several diagrams to give a clear picture on the issue of false alarms. Individual Snort rules were examined to further analyse the false alarms issue and the impact of false alarms on IDS detection rate.

4 Results

Our earlier evaluation [21], which was conducted to focus on the issue of false alarms in real network traffic, asserted that the problem remains critical for current detection systems. Hence, this experiment was carried out to endorse our previous findings by highlighting the issue of the false alarm rate on the DARPA dataset.

Snort has generated a total of 91,671 alerts, triggered by 115 signature rules, in this experiment. Of the alerts generated from this dataset, around 63,000 (69%) were false positives. Significantly, this experiment had revealed a similar result to that yielded in our previous evaluation as well as Kayacik and Zincir-Heywood [11]. The false alarms have significantly outnumbered the true alarms.

To obtain a more in-depth understanding of the nature of Snort's alert generation, Figure 1 portrays a ROC plot [4] for the overall result, which illustrates the overall alert generation of Snort's signature rule. Since most plots have the value of X-axis and Y-axis less than 2000, Figure 2 depicts a clearer picture by focusing upon the area in coordinate 0 to 2000. The number of false positives generated is presented per signature for the X-scale, while the true positive is portrayed for the Y-scale. This diagram also describes the random guess line (non-discriminatory line), which gives a point along a diagonal line from the left bottom (0, 0) to the top right corner (10, 10). This diagonal line divides the space into two domains; namely good and bad figures. Ideally, a good detection system should generate a zero value for the X-scale; meaning no false positive has been generated by a particular signature rule. The area below the line represents a higher number of false positives than true positives. Thus, the more plots scattered on this area, the poorer the IDS is.

As the plot diagram can only give an overview of IDS alert generation, Figure 3 provides the exact figures of Snort's signatures generating the false and true positive alerts in a Venn diagram [18]. Surprisingly, 73 signatures had raised the false positive alarms; of which 26 of them had triggered both true and false positives. It is also worth noticing that of those 26 rules, 14 signatures had false positives outnumbering the true positives. This seems to be a very critical issue faced by contemporary IDSs. The following subsections discuss this issue in greater detail.

4.1 True Positive

Given that the objective of this experiment is to investigate the issue of IDS false alarms, evaluating Snort's detection performance on DARPA dataset is beyond the scope of this study. In this paper, therefore, we will not further evaluate the extent of Snort's detection performance on a particular attack in a greater

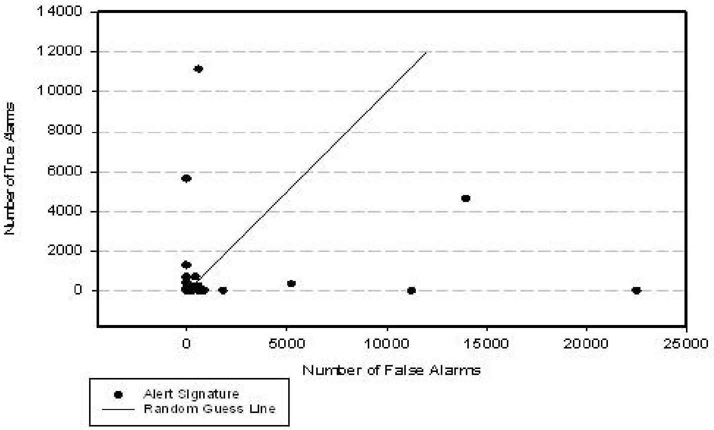


Fig. 1. Overall Alert Generation per Signature

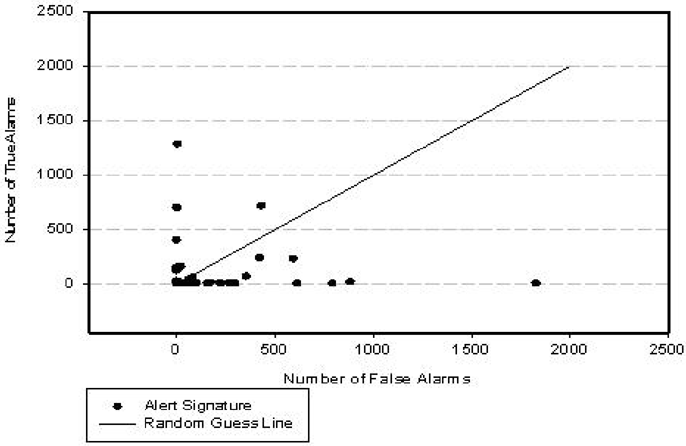


Fig. 2. Alert Generation per Signature within Cartesian Coordinate (2000, 2000)

detail. However, this subsection presents a brief overview of the Snort detection performance on 4 attack categories, namely probe, Denial of Services (DoS), Remote to Local (R2L) and User to Root (U2R).

In this experiment, 42 of the total 115 signatures had generated pure true positives. Approximately only 31% (27,982 alerts) of total alerts generated by 68 signatures were asserted as true positives. Interestingly, about 72% of them were generated due to the probing activities.

Generally, Snort fares well in detecting probe attacks, which largely generate noisy connections. In this study, we found that Snort has a very low threshold for detecting probing activity; for example in detecting ICMP connections. This had made up of 40% (37,322 alerts) of the total alerts. In spite of its sensitivity, Snort had generated a small number of true ICMP alarms in this experiment, which accounted for only 13% of those 37,322 alerts. This significantly highlights the underlying flaw of Snort IDS alarms.

In term of the DoS attacks, Snort did not perform well. Only one attack, named Back [12], was detected without generating any false positives. This had contributed to 20% of total true alarms. As for remote to local (R2L) attacks, about 16 out of 20 types of attacks had been detected. This, however, only

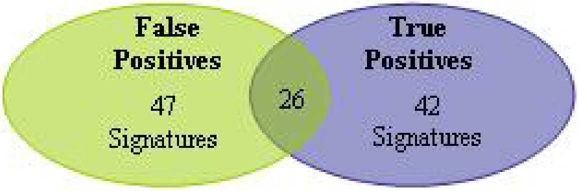


Fig. 3. Snort IDS Alarms - True and False Positives

made up of 2% of true alarms. Although Snort fares well in this category, it had critically missed numerous attack instances, such as "ppmacro" attack [12].

The last attack category, user to root (U2R), is the most challenging attack for Snort IDS. Since U2R attack typically occurs on a local machine, which attempts to elevate administrator's privileges, it relies solely on a system log or computer's filesystem. As such, Snort, a network-based IDS that merely depends on network connections, does not work well in detecting such attacks. Indeed, only a small proportion of true alerts (less than 1%) were generated owing to this category.

Overall, the experiment has yielded a similar result as the one revealed by Brugger and Chow [8]. Snort's performance does not look particularly impressive. Although there were quite a significant number of true alarms (27,982 alerts), only 32 from 54 types of attacks were detected. In fact, not all instances from those 32 attacks were perfectly detected by Snort. This emphasises the fact that Snort was not designed to detect all types of attacks, but rather to work in conjunction with other IDSs to achieve the best detection performance.

4.2 False Positive

Approximately, 69% of total alarms are affirmed to be false positives. Figure 4 shows the top 5 false alarms raised by Snort. Interestingly, 48% of the total false alarms were made up of ICMP alerts. Logging every connection associated with probing, for example all ping activities, will only generate a huge number of false positives. In fact, all detected ICMP traffic did not surely imply the occurrence of probing actions, but it was merely an informational event, which possibly indicates the occurrence of network outage.

In term of the alert categories, 39% (24,835 alerts) of the total false alerts were triggered due to a policy violation. Significantly, these types of alerts are more related to irrelevant positives than false positives. Irrelevant positives refer to the alerts generated from unsuccessful attempts or unrelated vulnerability, which do not require immediate actions from the administrators. However, as those informational alerts were not related to any suspicious activity from DARPA attack database and in order to make it simpler, they will be referred to as false positives.

The highest number of false alarms in this experiment was triggered by INFO web bug 1x1 gif attempt signature. This signature rule was raised when the privacy policy violation was detected [20]. Theoretically, the web bug is a graphic on the web page and email message, which is used to monitor users' behaviours. This is often invisible (typically only 1x1 pixel in size) and hidden to conceal the fact that the surveillance is taking place [19]. In fact, it is also possible to place web bug in a Word document as it allows html in a document or images to be downloaded from the external server. This is particularly useful if the document is supposed to be kept private, and web bug provides the information if the document had leaked by finding out how many IP addresses had looked at it [1]. Since none of these web bug alerts related to any attack instances, our study reveals that no true alarms associated with this signature had been generated. Therefore, 22,559 alerts from this signature were entirely asserted as false positives. This contributed to 35% of the total false alarms raised by the system. Although both ICMP and web-bug alerts can be easily filtered by the administrator through disabling or altering the signature rules, simply tuning the Snort signatures could increase the risk of missing real attacks.

Another similar policy-related alarm logged in this experiment is CHAT IRC alerts. These alerts accounted for 3.6% (2,276 alerts) of the total false alarms. Snort generates these IRC alerts because the network chat clients have been detected. In common with the previous "web bug" signature, IRC alerts were not truly false positives. Principally, Snort, given the correct rule, fares well in detecting policy violation. Indeed, through the investigation of the DARPA packet payload, it was noticeable that the chat activity did take place on a certain time. However, since these alerts did not contribute to any attack instances in the attack list, we would assume these as false positives. These CHAT IRC alerts were triggered by 3 signature rules; namely CHAT IRC message, CHAT IRC nick change and CHAT IRC channel join.

Apart from those top 5 false alarms signatures shown in Figure 4, there were 68 other signatures that generated false alarms. Of the total 115 signatures, 47 of them had triggered one hundred per cent false positives. All these alerts are known as pure false positive alarms since they are not in common with any true alarms. Significantly, 25 of those 47 signatures were web-related signatures. Although port 80 was one of the most vulnerable ports for DARPA attacks, these signatures did not correspond to any attack instances listed in the attack database. The effectiveness of Snort rules in detecting web-related attacks largely hinges on the effectiveness of keyword spotting. Most of the rules looking for web-related attacks are loosely written and merely checked on the presence of a particular string in the packet payload. This renders the system prone generating a superfluous number of false alerts. Aside from the web-related alerts, other 22 signatures, involving ICMP informational rule, policy, preprocessors, exploit attempt and SQL rules, had also generated a significant number of false positives, which accounted for 44% (28340 alerts) of total false alarms raised by the system.

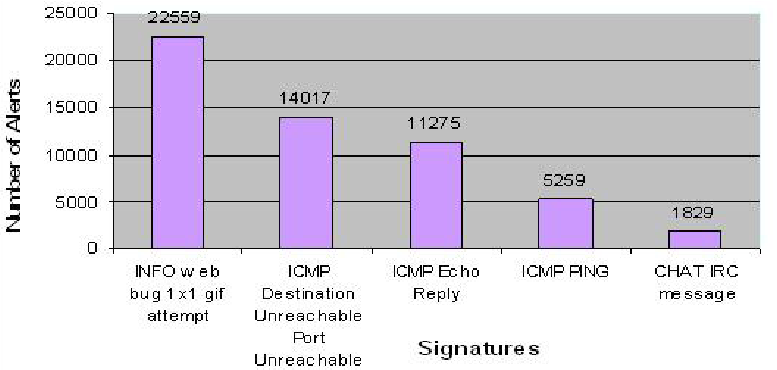


Fig. 4. Top 5 False Alarms

Despite the informational and policy-related alerts, the pure false positives could also be generated due to the loosely written rules of Snort IDS. For example, the vulnerability of Snort in relying on the keyword spotting is intolerable. This has been further discussed in Tjhai et al [21].

As described earlier, exact 14 signatures has produced more false positives than true positives. This highlights the critical issue of false alarms in the real world. The process of identifying the real attacks could be undermined if the false positives per signature highly outnumbered the true positives. In addition, this could render the administrator apathetic; thus tending to conclude that any such alerts as false positives. As a consequence, this problem could seriously inhibit IDS detection performance in a real environment.

While Snort could detect 32 types of attacks, it had produced a large volume of unnecessary alerts; in term of its alerts' quality. One of the good examples can be taken from this experiment is the alerts triggered due to the detection of "Back" DoS attack, by WEB-MISC apache directory disclosure attempt signature. Only 7 instances from this attack were included into the DARPA dataset, but surprisingly Snort detected all 7 instances by triggering 5,628 alerts from single signature. Obviously, Snort has generated a huge number of redundant alerts in this case. Indeed, this often leaves the administrator with the difficulty of verifying every single alert.

In term of the false positives, the experiment revealed a slightly different result as generated by Brugger and Chow [8]. A smaller number of false alarms, accounted for 11% of total alerts, had been estimated by Brugger and Chow, compared to our result (69% of total alerts). The insignificant number of false alarms was presumably due to the removal of "web-bugs" rule that had generated a very significant number of false alarms. This signature was believed to not provide any true positives, and could potentially prevent an objective evaluation of false positives produced by an IDS. As for Snort rules, only 36 signatures were triggered in their study. However, the "ICMP Destination Unreachable Port Unreachable" signature had produced the second highest number of false alarms, similar to our result.

The experiment result has shown that Snort has produced an unacceptable number of false alarms. However, the evaluation of 18 IDSs on 1999 DARPA dataset, had yielded a remarkable result, indicating that most systems had false alarm rates which were low and well below 10 false alarms per day [14]. This might be due to their ability to tune the systems to reduce the false alarms on three weeks of training data prior to running the two weeks of test data.

5 Conclusions

Given the time span between the creation of DARPA dataset and the Snort rules, we initially thought that Snort could fare well in detecting DARPA attacks. What we found instead was that the detection performance was low; only 32 attacks were detected, and Snort has produced a large volume of false positives. Indeed, not all instances of those 32 attacks have been perfectly detected by

Snort. From this experiment, it is obvious that the issue of false alarm has become very critical. The false positives outnumbered the true positive by a ratio of 2:1. In fact, more than half of the signatures producing both true and false positives in this evaluation have triggered more false positive than true positive alarms. This issue would critically reduce IDS detection performance; not only in this simulated network environment but also in a real environment.

Regarding the quality of alerts generated, Snort generated a huge number of redundant alerts, which critically highlighted the performance issue of the Snort alert reporting system. This often leaves the administrator with overwhelming alerts, which renders alert validation difficult to manage. Importantly, this issue has also driven the need to have an improved or better alarm reporting system through the implementation of alarm suppression and correlation methods.

Apart from total 39,849 false alerts triggered by 12 signatures generating both false and true alarms, Snort has also produced 28,340 pure false positive alarms, which can be arguably expected to happen in a real-network environment. Interestingly, this has accounted for 31% of alarms. However, in this experiment, we have not had a chance to individually track the cause of these alerts. Having said that, we believe that this might be caused by the nature of Snort IDS, which relies on keyword spotting (i.e. matching the packet content to signature rule) to detect malicious activity. Significantly, this finding underlines another weakness of Snort IDS, which could render the system prone to produce excessive alerts.

Overall, our study has confirmed the criticality of the IDS false alarm issue. Given the findings in this evaluation, endorsed by our previous experimental results, it is clear that false alarm is a never-ending issue faced by current IDS. The sensitivity of Snort rules in detecting probing activities can generate a large volume of false positives.

The ideal solutions to this problem is either to tune the IDS signature rules; this should be done by a skillful administrator who has the knowledge of security and knows well the environment of the protected network, or alternatively to focus upon the alarm correlation, which aims to improve the quality of the alerts generated. The idea of reducing false alarm in alarm correlation system has become the main subject of current IDS research. However, apart from the false alarm reduction, the alert management or alert correlation should also be aimed at the presentation of the IDS alerts itself to the system administrator. This might include the reduction of the redundant alerts and the aggregation of the related alerts (i.e. various alerts generated by a single attack).

References

1. Adoko, What Are Web Bugs? (2008) (date visited: September 7, 2007), http://www.adoko.com/webbugs.html
2. Alharby, A., Imai, H.: IDS False alarm reduction using continuous and discontinuous patterns. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 192–205. Springer, Heidelberg (2005)

3. Alshammari, R., Sonamthiang, S., Teimouri, M., Riordan, D.: Using Neuro-Fuzzy Approach to Reduce False Positive Alerts. In: Communication Networks and Services Research, 2007. Fifth Annual Conference. CNSR 2007, pp. 345–349 (2007)
4. Anaesthetist, The magnificent ROC (2007) (date visited: August 17, 2007), http://www.anaesthetist.com/mnm/stats/roc/Findex.htm
5. Axelsson, S.: The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM Transactions on Information and System Security 3(3), 186–205 (2000) (date visited: May 10, 2007), http://www.scs.carleton.ca/ soma/id-2007w/readings/axelsson-base-rate.pdf
6. BASE, Basic Analysis and Security Engine (BASE) Project (2007) (date visited: April 25, 2007), http://base.secureideas.net/
7. Bolzoni, D. and Etalle, S.: APHRODITE: an Anomaly-based Architecture for False Positive Reduction (2006) (date visited: November 7, 2006), http://arxiv.org/PScache/cs/pdf/0604/0604026.pdf
8. Brugger, S. T. and Chow, J.: An Assessment of the DARPA IDS Evaluation Dataset Using Snort (2005) (date visited: May 2, 2007), http://www.cs.ucdavis.edu/research/tech-reports/2007/CSE-2007-1.pdf
9. Caswell, B., Roesch, M.: Snort: The open source network intrusion detection system (2004) (date visited: October 3, 2006), http://www.snort.org/
10. Julisch, K.: Mining Alarm Clusters to Improve Alarm Handling Efficiency. In: Proceedings of the 17th Annual Conference on Computer Security Applications, pp. 12–21 (2001)
11. Kayacik, G.H., Zincir-Heywood, A.N.: Using Intrusion Detection Systems with a Firewall: Evaluation on DARPA 99 Dataset. NIMS Technical Report #062003 (June 2003) (date visited: September 9, 2007), http://projects.cs.dal.ca/projectx/files/NIMS06-2003.pdf
12. Lincoln Lab, DARPA Intrusion Detection Evaluation (2001) (date visited: May 15, 2007), http://www.ll.mit.edu/IST/ideval/
13. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K., Zissman, M.A.: Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. In: Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX) (1999) (date visited: July 8, 2007), http://www.ll.mit.edu/IST/ideval/pubs/2000/discex00_paper.pdf
14. Lippmann, R.P., Haines, J.W., Fried, D.J., Korba, J., Das, K.J.: The 1999 DARPA off-line intrusion detection evaluation. Computer Networks 34, 579–595 (2000) (date visited: June 20, 2007), http://ngi.ll.mit.edu/IST/ideval/pubs/2000/1999Eval-ComputerNetworks2000.pdf
15. Mahoney, M.V., Chan, P.K.: An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. In: Vigna, G., Krügel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 220–237. Springer, Heidelberg (2003) (date visited: June 22, 2007), http://cs.fit.edu/~mmahoney/paper7.pdf
16. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans. Information System Security 3(4), 262–294 (2000) (date visited: June 19, 2007), http://www.cc.gatech.edu/~wenke/ids-readings/mchugh_ll_critique.pdf
17. Roesch, M.: Snort - Lightweight Intrusion Detection for Networks. In: Proceedings of LISA 1999: 13th Systems Administration Conference, Seattle, Washington, USA, November 7-12 (1999)

18. Ruskey, F., Weston, M.: A Survey of Venn Diagrams (2005) (date visited: October 10, 2007), http://www.combinatorics.org/Surveys/ds5/VennEJC.html
19. Smith, R.: The Web Bug FAQ (1999) (date visited: August 15, 2007), http://w2.eff.org/Privacy/Marketing/web_bug.html
20. Snort, INFO web bug 1x1 gif attempt (2007) (date visited: August 9, 2007), http://snort.org/pub-bin/sigs.cgi?sid=2925
21. Tjhai, G., Papadaki, M., Furnell, S., Clarke, N.: Investigating the problem of IDS false alarms: An experimental study using Snort. In: IFIP SEC 2008, Milan, Italy, September 8-10 (2008)

A Generic Intrusion Detection Game Model in IT Security

Ioanna Kantzavelou[1] and Sokratis Katsikas[2]

[1] Dept. of Information and Communication Systems Engineering
University of the Aegean, GR-83200 Karlovassi, Samos, Greece
ikantz@aegean.gr
[2] Dept. of Technology Education and Digital Systems
University of Piraeus, 150 Androutsou St., GR-18532 Pireaus, Greece
ska@unipi.gr

Abstract. Intrusion Detection has a central role in every organization's IT Security. However, limitations and problems prevent the commercial spread of Intrusion Detection Systems. This paper presents an attempt to improve Intrusion Detection benefits with the use of Game Theory. A generic intrusion detection game model that reveals the way an IDS interacts with a user is described and examined thoroughly. Moreover, a specific scenario with an internal attacker and an IDS is presented in a normal form game to validate the functioning of the proposed model. Solutions for this game are given as a one shot game as well as an infinitely repeated game.

Keywords: Intrusion Detection, noncooperative Game Theory, internal attacker.

1 Introduction

Peter Denning argues that Computer Science is being always expanding, whenever it forms a new relation with another field, generating a new field [1]. As an example, Intrusion Detection has incorporated a large variety of different tools, methods, and techniques until today. However, the same problems torture this highly demanded field without significant progress. Therefore, it is necessary to redirect research to new fields of science with potential to give solutions.

The area of Intrusion Detection includes the meanings of monitoring and decision. Intrusion Detection is the monitoring of a system's events and the decision whether an event is *normal* or *abnormal*. The word *normal* defines every event that is consistent with the security policy applied to the system, and the word *abnormal* defines any event that threatens the security status of the system. Besides, as IT is a human-computer interactive situation, Intrusion Detection in IT security is also an interactive situation.

"Game Theory is that branch of decision theory which deals with the case in which decision problems interact" [2]. Many disciplines have incorporated game theoretic techniques, including Economics, Law, Biology, Psychology and Political Philosophy. The similarities between the two fields of Intrusion Detection and

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 151–162, 2008.

Game Theory, deserve further study and research. Therefore, the application of Game Theory to Intrusion Detection will show how much this new approach might improve Intrusion Detection's real positive outcomes.

In this paper, we present a generic Intrusion Detection game model between a user and an IDS in extensive form. The description of this model reveals its elements, how it is played, and how it could be integrated as a mechanism in an IDS. Because it is a repeated game, the iterated parts of it are separated in order to define its structure. Following this, a two-player noncooperative game between an internal attacker and an IDS is constructed, discussed, and solved to illustrate the model's functionality and implementation feasibility. Because the model is generic, its flexibility allows the implementation of several different cases and the construction of an entire game-based Intrusion Detection mechanism to be incorporated in an IDS.

The remainder of this paper is organized as follows. Section 2, discusses related work in this field. Consequently, Intrusion Detection is represented in a generic game model in Sect. 3, by identifying players, specifying their actions and preferences over them. Then, in Sect. 4, a strategic form game between an IDS and an internal attacker is modeled and solved as a static and as a repeated game too. Finally, we review the results of our work and highlight future steps for the continuation of our research, in Sect. 5.

2 Related Work

Interesting attempts, to formulate Intrusion Detection as a game, have been recently appear in the literature. Remarkable conclusions for the motivation of using game theory in Intrusion Detection are presented in [3]. Cavusoglou et. al. use game theory to configure the result derived from a classical IDS [4]. Similarly, Alpcan et. al. present an incomplete information game for the verification of the IDS's output in [5] and [6]. These approaches aim at using game theory over the output of an IDS, rather than for detection. In other words, they assume that an IDS first detects using classical techniques and approaches, and subsequently their approach operates at a second stage to assist detection and optimize its results.

A general-sum stochastic game between an attacker and the administrator of a Target System has been constructed in [7], so that, this approach too does not use Game Theory for a straightforward detection. A similar problem, the problem of detecting an intruding packet in a communication network has been considered in [8]. The described game theoretic framework has been formulated in such a way, that the intruder picks paths to minimize chances of detection, whereas the network operator chooses a sampling strategy - among the developed sampling schemes - to maximize the chances of detection. In these approaches, the games are constructed between a person (administrator, network operator) and a user, not between an IDS and a user.

In [9] and [10] Intrusion Detection is modeled as an incomplete information game, a basic signaling game, for mobile ad hoc networks. This work is very close to [5] and [6]. Finally, Agah et. al. describe a repeated game for preventing DoS attacks in wireless sensor networks by proposing a protocol between an intrusion detector and the nodes of a sensor network [11].

Our approach addresses problems in the area of Intrusion Detection, and presents a generic intrusion detection game model, to identify the interactions between a user and an IDS. In particular, we determine how a user will interact in the future, and how an IDS will react to protect the Target System it monitors. Unlike the described related work, our model is generic with an open well defined structure that allows its application to many real life cases.

3 A Generic Game Model between a User and an IDS

Because of the dynamic nature in the interactions between a user and an IDS and because game theory models dynamic interactions using the extensive form representation [12], we present the generic game model of intrusion detection as an extensive form game.

3.1 Players

To formulate an extensive form game between an Intrusion Detection System (IDS) and a user who is intending to use a Target System (TS) that is behind this IDS, five elements must be specified: the players, their possible actions, what the players know when they act, a specification of how the players' actions lead to outcomes, and a specification of the players' preferences over outcomes. This game has two players, an IDS called I and a user called U. The player U might be a normal user or an attacker and even if he is a normal user he might unintentionally harm the TS. Therefore, the player U is considered as a general user and no further categorization is needed before he acts.

3.2 Actions/Moves

Each player has a number of possible actions to perform. Player I, examining player's U actions, allow player U to continue using the TS by choosing C, if player I comes to the conclusion that player U acts legitimately. Conversely, player I chooses P to prevent additional damage to the TS, if it decides that player U is doing illegal actions. In short, in this game player I has two choices; choice C to allow player U to continue using the TS and choice P to prevent player U to attack or to further damage the TS. In real cases, this binary approach reflects that player U requests a service or a resource from the TS, and player I either accepts to fulfil the request (choice C) or refuses it (choice P). Although other approaches might appear to include more than two choices (see Sect.4), the interpretation is the same.

Similarly, player U has three possible actions; L when acting legitimately, A when acting illegally generating attacks, and E when he decides to exit the TS and so he logs out. Comparing to player's I actions, player U has one more action to choose, that is, he has three actions.

3.3 Sequential and Simultaneous Moves

Next, the key question to be answered for this game is how this game is being played, with simultaneous or with sequential moves. The crucial criterion to answer this question is to take into account first, that using a TS and requesting a service from it, the user waits for a response although most of the times he does not even realize it, and afterwards, the user makes another request to which the TS replies too, and so on. Thus, the kind of interaction formulated here is a sequential-move interaction, like the one taking place between two players in a chess game.

However, when the TS is protected by an IDS, the user is interacting with the TS but he is also interacting with the IDS. In the last kind of interaction, the user is acting and at the same time the IDS collects data related to this action, filters it and decides to counteract in the case of an attack. The IDS performs a counteraction ignoring the user's action at that instant.

Elaborating the described interaction into the game theoretical approach [13], in Intrusion Detection an attacker plans his moves before he acts and calculates the *future consequences*. Up to this point the game is a sequential-move game, but when the attacker starts applying this plan and confronts the existence of an IDS protecting the TS of his attack, then he is trying to discover what this IDS is going to do *right now*. The last reveals that the game includes also simultaneous moves.

On the contrary, an IDS has been designed and implemented incorporating one or more ID techniques which lead to a predefined plan of its moves, calculating the future consequences to the TS that protects. Up to this point again the game is for a second time a sequential-move game, but when a user enters the system, the IDS observes his moves to decide whether he is an attacker or not, and according to its design, to figure out what the attacker is going to do right now. The conclusion once more is that the game includes also simultaneous moves. Consequently, Intrusion Detection in IT Security is a game that combines both sequential and simultaneous moves.

3.4 General Formal Description of the Game

Consider the extensive form of the Intrusion Detection game depicted in Fig. 1. A user (player U) attempts to enter a TS protected by an IDS (player I). The user might successfully login to the TS or not (e.g. typing in a wrong password). Even if he gains access to the TS he might be already member of a black list. Therefore, player I moves first at the initial node (the root) of this game denoted by an open circle, when player U attempts to enter the TS. Then, examining this attempt, player I has two choices; to allow user continuing (choice C) or to prevent user from using the TS (choice P) which ends the game. In the latter case, it is assumed that player I has achieved to detect a real potential attacker and has not caused a false alarm. Hence, the outcome at this end of the game is the vector *(detection, attempt of an attack)* for player I and player U respectively.

If choice C has been selected by player I, then player U has three choices; to perform legal actions (choice L), to attack the TS (choice A), or to exit from the

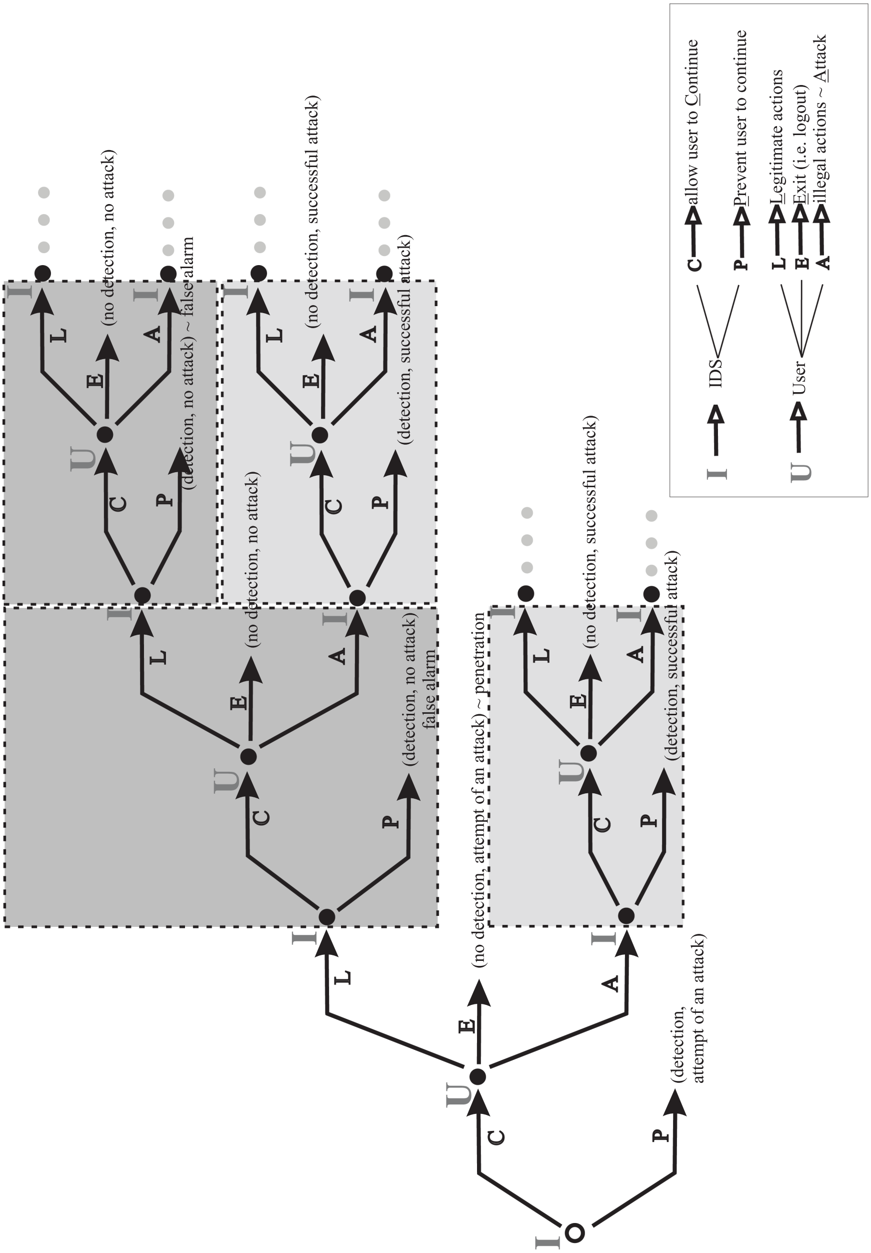

I
C
P
U
(detection, attempt of an attack)
L
E
A
(no detection, attempt of an attack) ~ penetration
(no detection, no attack)
(detection, no attack) false alarm
(no detection, successful attack)
(detection, successful attack)
(detection, no attack) ~ false alarm
I
U
C allow user to Continue
P Prevent user to continue
IDS
User
L Legitimate actions
E Exit (i.e. logout)
A illegal actions ~ Attack

Fig. 1. Intrusion Detection as an extensive form game

TS (choice E). If player U exits the TS, then the game ends with outcomes *(no detection, attempt of an attack)*. The reason for these payoffs is first that the user has achieved to enter the TS and the IDS did not detect an attack even if he was a masquerade, so this is counted as penetration, and second that the user did not attack the TS although he might got the keys (the pair username and password) and checked them against the TS, to attack it another time in the future.

The game continues with player U selecting a legal action over the TS or attacking the TS. In both instances, player I analyzes afterwards the opponent's move, and decides whether he is acting legally or not. If player I chooses P, then the payoffs of the game totally diverge. In particular, if player U has committed an attack (choice A) then the payoffs are *(detection, successful attack)*, otherwise (choice L), the payoffs are *(detection, no attack)* which constitutes a false alarm.

Alternatively, when player I allows player U continuing working with the TS (choice C), while player U is acting legally, then player U might either proceed with legal actions (choice L), or with an attack (choice A), or he decides to exit the TS (choice E) terminating the game. This outcome of the game results in the payoffs *(no detection, no attack)*. The described stage of the game surrounded by a dashed line rectangle as shown in Fig. 1, is a repeated division of the game which leads to the end of the game when player U chooses E.

Exploring further the extensive form of the ID game for repeated divisions, we locate two parts; the one is related to legal actions and the other one to attacks. Figure 2 represents the extensive form game explained in detail above, displaying two separate divisions and their iterations. Although the form of the ID game looks as never ending, each of the repeated divisions definitely has a branch where the game ends, and thus the game under study is a finite game.

3.5 Checking the Extensive Form

Extensive form games should give a picture of a tree. There are two rules which ensure this form [12]; first, the number of arrows pointing out from a node must be at least one, and the number of arrows pointing at a node must be at most one, and second, retracing the tree in a backward fashion from a node towards the initial node, the starting node should not be reached again drawing a cycle, but actually the initial node should be the end of this backtracking.

The first rule implies that a player has at least one action to perform when it is his turn to play, and that after an action of a player, either another player is next, or the game ends to a payoff vector of a specific outcome. The second rule aims at solving games in extensive form using backward induction, since they have the form of a tree.

The Intrusion Detection game described above has been modeled in the form of a tree. Checking the game against the first rule, it is apparent that the number of arrows pointing in any node, as well as the number of arrows pointing out from any node, satisfy this rule. Similarly, examining the plausibility of backtracking from any node towards the initial node of the game, no circle would be drawn and the initial node would be reached.

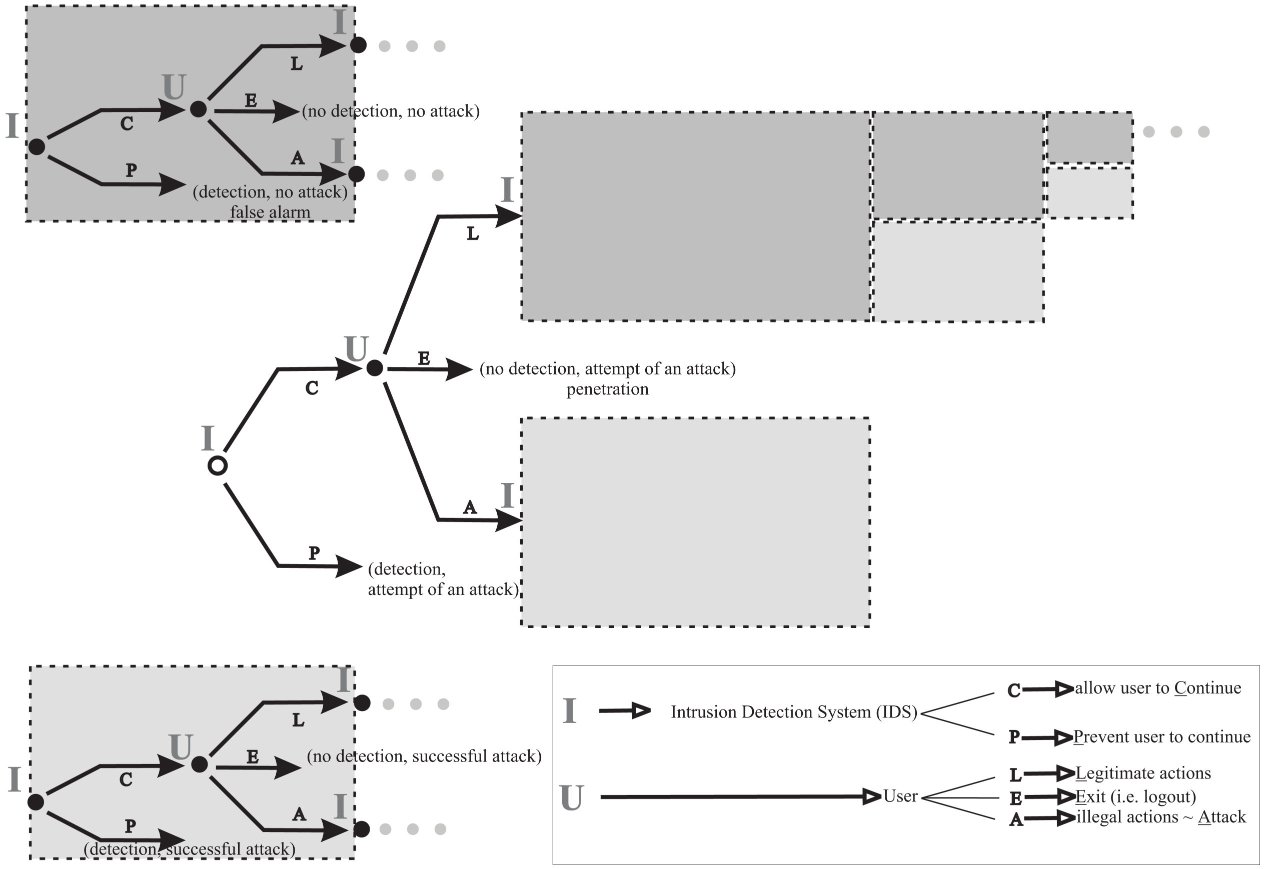

I
C
P
U
L
E
A
I
I
(no detection, no attack)
(detection, no attack)
false alarm
I
C
P
U
L
E
A
I
I
(no detection, attempt of an attack)
penetration
(detection,
attempt of an attack)
I
C
P
U
L
E
A
I
I
(no detection, successful attack)
(detection, successful attack)
I
Intrusion Detection System (IDS)
C allow user to Continue
P Prevent user to continue
U
User
L Legitimate actions
E Exit (i.e. logout)
A illegal actions ~ Attack

Fig. 2. Intrusion Detection and the repeated divisions of the game

4 Playing the Game with an Internal Attacker

To validate the game model presented in Sect. 3, we constructed a game to be played between an IDS and a user of the TS behind this IDS. In addition, we assumed that the user is an internal attacker too, also called an *insider*. It is hard to detect insiders because of their privileges that partially protect them from being caught. Such an endeavor generates a great number of false positive alarms, with an analogous number of false negative alarms.

The game has two players, the internal attacker U and the IDS I. Although for purposes of simplicity and appearance, the generic game model implies a black and white representation, one of the things we examined was the action set of each player. It seems that, an internal attacker of an organization acts normally according to his commitments and duties, occasionally he makes mistakes, he also acts methodologically to prepare attacks, and finally he executes attacks as planned. The IDS follows a direction towards cooperation and decides among four alternatives. First, it allows the user to continue if nothing suspicious has been noticed, second it makes a recommendation whenever slight deviations are encountered, third it raises a warning to remind the user to be consistent with their agreement on the regulations of using the TS, and fourth it stops the user when a violation is detected.

Summarizing, the player U has four strategies, Normal, Mistake, Pre-Attack, Attack, and the player IDS has another set of four strategies, Continue, Recommend, Warning, Stop. Obviously, for the player U the first strategy N corresponds to a legal action while the rest of strategies are equivalent to an illegal action. Likewise, for the player$I$ the first three strategies correspond to a permission to continue, whereas the latter one, the S is equivalent to prevent. Transferring this game from the extensive form of the game model to a strategic form, we get the following 4x4 matrix, as presented in Table 1. The row player is the user (U) and the column player the IDS (I).

The outcomes of the game have been quantified, by first specifying preferences over outcomes, and then by using the von Neumann-Morgenstern utility function.

In particular, ranking user's preferences over strategies, from the least preferable (PS) to the most preferable one (AC), we get the following:

$$PS_U \prec MS_U \sim PW_U \prec PR_U \sim MW_U \prec MR_U \sim AR_U \prec AW_U \sim NS_U \prec$$
$$AS_U \sim NW_U \prec NR_U \prec MC_U \prec PC_U \prec NC_U \prec AC_U$$

Next, we assigned numbers to reflect these preferences, by defining the von Neumann-Morgenstern utility function for player user. We set 0 to strategy PS and 1 to strategy AC. Using rational numbers we assigned a value to every strategy according to the above ranking. Then we got the values free of fractions, after multiplying with their least common factor. The user's payoffs are given as the first number in each pair of outcomes, as displayed in Table 1.

Table 1. A game between an internal attacker and an IDS in normal form

		I D S			
		C	**R**	**W**	**S**
U	**N**	(13,17)	(9,5)	(8,4)	(7,2)
	M	(10,3)	(6,6)	(5,7)	(4,14)
	P	(12,1)	(5,8)	(4,9)	(3,15)
	A	(19,0)	(6,10)	(7,11)	(8,16)

Following the same steps to rank the IDS's preferences, we establish the following:

$$AC_{IDS} \prec PC_{IDS} \prec NS_{IDS} \prec MC_{IDS} \prec NW_{IDS} \prec NR_{IDS} \prec MR_{IDS} \prec MW_{IDS} \prec PR_{IDS} \prec PW_{IDS} \prec AR_{IDS} \prec AW_{IDS} \prec MS_{IDS} \prec PS_{IDS} \prec AS_{IDS} \prec NC_{IDS}$$

Finally, assigning numbers to quantify the relations between IDS's preferences, we resulted in the payoffs also shown in Table 1 as the second number in each pair of outcomes.

4.1 Solving the Game

Having described Intrusion Detection as a 2-player noncooperative game, we proceed to solve it using equilibrium analysis. An equilibrium in a game is a set of players' decisions that results in an outcome, such that, no player has any reason to deviate from his choices, given that all the players do the same. John Nash proved that every noncooperative game has at least one Nash equilibrium (NE). The notion of NE is the most commonly used solution concept in noncooperative game theory [2].

We located the solution of this game by examining players' best responses. There is a unique Nash Equilibrium (NE) which corresponds to the strategy profile (combination) AS with payoffs (8,16). Moreover, we used the Gambit tool [14] to verify our solution and it calculated the same result. It is a perfect NE that reveals the intention of the internal attacker to attack the system and the reaction of the IDS to stop him doing so.

Interestingly, there is another pair of strategies, the NC strategy profile, with payoffs (13,17) which are both greater than the corresponding of the NE. Besides, payoffs (13,17) are absolutely the highest each player can get in this game. In fact, strategy NC Pareto dominates[1] the NE, and because the corresponding payoffs are the highest, the NC strategy is Pareto efficient[2]. In other words, any player between the two can increase his outcome by deviating from the equilibrium path, that is the strategy AS, and choose the Pareto efficient dominant strategy NC.

[1] A strategy Pareto dominates another strategy if the outcome of the first is higher than the outcome of the latter one (Vilfredo Pareto, 1848-1923).

[2] A strategy is Pareto efficient if no other strategy yields all players higher payoffs [15].

In a one shot game as this has been presented in the normal form, real circumstances cannot be depicted nor examined in depth, to find realistic solutions. Indeed, especially in this game, the players play the game repeatedly infinite number of times. The reason is that, the user is not a random attacker, but an internal user of the system, who spends a long time every day in front of it. In the generic game model described in Sect. 3, parts of repeated divisions have already been located.

To solve this game as a repeated game we followed D. K. Levine's step-by-step procedure, as described in [16]. First, we decided to use the average present value method to aggregate payoffs obtained among different periods. Alternatives would have been to add payoffs together, to average them, or to take the present value. Second, we clarified that this game can be repeated infinite number of times, as mentioned in the previous paragraph. Finally, regarding the discount factor δ that shows how much impatient a player is, we defined a common discount factor for both players. The discount factor δ varies between zero and one, with zero to match an impatient player and one a patient one. In our case, the internal attacker is a patient player, because he has plenty of time to organize and execute an attack. In addition, the IDS is inherently a patient player, because it plays infinitely with a user of the TS, although it does not know that he is an internal attacker.

One of the things we examined is the grim strategies equilibria. Considering a case where the internal attacker plays a certain strategy at every period, we examined the circumstances under which he would deviate from this equilibrium path, and he would decide to play another strategy. But the IDS should react by choosing a 'punishment' strategy against such a deviation, known as grim strategy. We looked specifically at the following scenario:

Case 1. The internal attacker starts playing the game acting legitimately, by choosing strategy *N*. The IDS corresponds by playing strategy *C*. This goes on at every period as long as NC strategies are being played. But then, under which circumstances the internal attacker will commit an attack, i.e. deviate from this equilibrium path? We are looking for the discount factor δ that will motivate the internal attacker to do so because his benefit will be higher.

Solution: We calculate the average present value (APV) for each player on the equilibrium path using the following formula and the related identity,

$$(1-\delta)\cdot\left(u_1+\delta u_2+\delta^2 u_3+\delta^3 u_4+\ldots\right). \tag{1}$$

where $u_i, i = 1, 2, 3...$, is the payoff a player receives at period i and δ is the common discount factor.

$$1+\delta+\delta^2+\delta^3+\ldots=\frac{1}{1-\delta} \tag{2}$$

We found $APV_U = 13$ and $APV_I = 17$ respectively as expected, because same strategies are being played at every period. Following this, we examined players' best responses when the opponent follows the selected equilibrium path.

When the internal attacker follows the strategy N, then the IDS's best response is strategy C. But when the IDS follows the strategy C, then the internal attacker's best response is strategy A because his highest payoff is 19. Next, we calculated the APV for each player if each follows the equilibrium path at the first period but then they both deviate and continue playing the static NE as calculated. The results are $APV_I = 17 - \delta$ and $APV_U = 19 - 11 \cdot \delta$.

Finally, we compared the average present value to remaining on the equilibrium path with that to deviating. In other words, we determined the discount factor δ for which a player will stick with his first choice and will not change afterwards by attacking the TS. The δ discount factor must be greater or equal to $\frac{6}{11}$ which is reasonable for this type of attacker. The fact is that, the internal attacker behaves as a patient player when he takes his time to complete an attack but he is impatient enough when time pushes him to finish with his illegal activities. The result can be verified by calculating the limit of the derived average present values when δ is close to 1, that is to say, when players are patient.

$$\lim_{\delta \to 1} (17 - \delta) = 16 \tag{3}$$

$$\lim_{\delta \to 1} (19 - 11 \cdot \delta) = 8 \tag{4}$$

Apparently from (3) and (4), the derived formulas for the average present values for both players, give the payoffs of the static NE, when δ is close to 1.

5 Conclusions

A generic game model for the area of Intrusion Detection in IT security has been constructed, presented, and examined thoroughly. The game has been illustrated in an extensive form, and parts of repeated divisions have been located. It consists of both sequential and simultaneous moves, and follows the rules that ensure the extensive form of a game, and guarantees a solution of it. Subsequently, a snapshot of this generic game model has been isolated and a game between an internal attacker and an IDS is demonstrated and explained. We solved the game as a static game first and as an infinitely repeated game afterwards. The results show the potential of the implementation of such a framework within which an IDS and a user will safely interact preserving their own interests. In the future, a simulation of the proposed model should be set up to facilitate optimization of the described model and extension of the proposed approach to cover as many instances as possible.

References

1. Denning, P.: Is Computer Science Science? Communication of the ACM 48(4), 27–31 (2005)
2. Skyrms, B., Vanderschraaf, P.: Game theory. In: Gabbay, D.M., Smets, P. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 391–439. Kluwer Academic Publishers, Dordrecht (1998)

3. Ho, Y., Zhao, Q., Pepyne, D.: The No Free Lunch Theorems: Complexity and Security. IEEE Transactions on Automatic Control 48(5), 783–793 (2003)
4. Cavusoglu, H., Raghunathan, S.: Configuration of Intrusion Detection System: A Comparison of Decision and Game Theoretic Approaches. In: Proc. of the 24th International Conference on Information Systems, pp. 692–705 (December 2003)
5. Alpcan, T., Basar, T.: A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection. In: Proc. of the 42rd IEEE Conference on Decision and Control (CDC), Maki, HI, pp. 2595–2600 (December 2003)
6. Alpcan, T., Basar, T.: A Game Theoretic Analysis of Intrusion Detection in Access Control Systems. In: Proc. of the 43rd IEEE Conference on Decision and Control (CDC), Paradise Island, Bahamas, pp. 1568–1573 (December 2004)
7. Lye, K., Wing, J.: Game Strategies in Network Security. In: Proc. of the Foundations of Computer Security Workshop, Copenhagen, Denmark (July 2003)
8. Kodialam, M., Lakshman, T.: Detecting Network Intrusions via Sampling: A Game Theoretic Approach. In: Proc. of the IEEE INFOCOM 2003, San Fransisco (March 2003)
9. Patcha, A., Park, J.: A Game Theoretic Approach to Modeling Intrusion Detection in Mobile Ad Hoc Networks. In: Proc. of the 2004 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, pp. 280–284 (June 2004)
10. Patcha, A., Park, J.: A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks. International Journal of Network Security 2(2), 131–137 (2006)
11. Agah, A., Das, S.K.: Preventing DoS Attacks in Wireless Sensor Networks: A Repeated Game Theory Approach. International Journal of Network Security 5(2), 145–153 (2007)
12. Kreps, D.: Game Theory and Economic Modelling. Oxford University Press, Oxford (2003)
13. Dixit, A., Skeath, S.: Games of Strategy. W. W. Norton & Company, Inc. (1999)
14. McKelvey, R.D., McLennan, A.M., Turocy, T.L.: Gambit: Software Tools for Game Theory, version 0.2007.01.30 (January 2007) (accessed May 20, 2008), http://gambit.sourceforge.net
15. Osborne, M.J.: An Introduction to Game Theory. Oxford University Press, New York (2004)
16. Levine, D.K.: Repeated Games Step-by-Step (May 2002) (accessed March 1, 2008), http://www.dklevine.com/econ101/repeated-step.pfd

On the Design Dilemma in Dining Cryptographer Networks

Jens O. Oberender[1,⋆] and Hermann de Meer[1,2]

[1] Chair of Computer Networks and Computer Communications,
Faculty of Informatics and Mathematics
[2] Institute of IT-Security and Security Law,
University of Passau, Germany
{oberender,demeer}@uni-passau.de

Abstract. In a Dining Cryptographers network, the anonymity level raises with the number of participating users. This paper studies strategic behavior based on game theory. Strategic user behavior can cause sudden changes to the number of system participants and, in consequence, degrade anonymity. This is caused by system parameters that influence strategic behavior. Additionally, conflicting goals of participants result in dilemma games. Properties of message coding, e.g. collision robustness and disrupter identification, change the game outcome by preventing dilemmas and, therefore, enhance anonymity. Properties of anonymity metrics are proposed that allow for strategic user behavior.

1 Introduction

Anonymity systems protect the identities of communicating subjects disclosed. Beyond this coarse characterization, anonymity is seen as continuum – often related to a specified attacker model, e.g. an observer of network communications. Another definition of anonymity of a subject says, that the attacker cannot 'sufficiently' identify the subject within a set of subjects, the anonymity set [1]. The anonymity measure can be quantified as probability that the attacker correctly identifies a subject.

Because anonymity techniques are costly, users consider cost and benefit of anonymous communication, before they participate. In consequence, the design of anonymity systems must consider economically acting users. The benefit of participation in an anonymity system scales with the level of anonymity received. This paper identifies properties of anonymity measures necessary for strategic acting. Another open question is adjustment of design parameters for operating an anonymity system. The designer faces the dilemma, whether to maximize anonymity against powerful adversaries or minimize the cost of operation. The cost of countermeasures has an effect on the number of participating subjects, i.e. size of the anonymity set, which influences the level of anonymity.

[⋆] This work has been partly supported by the EuroFGI and EuroNF Networks of Excellence (IST-028022, 216366).

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 163–172, 2008.

Game theory is a branch of applied mathematics. It attempts mathematical modeling of strategic behavior. The objective of an iterative game is to maximize the average payoff per round. Nash Equilibria define game outcomes, in which none of the player can further increase its payoff. They are computed deterministically. Many participants in anonymity systems act strategically to enhance anonymity. In a DC-net, coding schemes enable identification of irrational adversaries (cf. Section 5). Therefore behavior of DC-net participants can be modeled using utility functions. These games can be studied using game theory. In practice, user preferences and aims of the adversary are unknown. Games with incomplete knowledge respect unknown strategies of other players.

This study evaluates behavior in a Dining Cryptographers network using a game theoretic model. The model considers properties of the coding schemes such as collision robustness and disrupter identification, and the anonymity preference of a user. The designer can apply an efficient and a collision robust design, the user participate or leave, and the adversary can disrupt or conform. We evaluate the Nash Equilibria and identify design parameters, where the level of anonymity is sufficiently high, users participate because of reasonable cost, and an adversary has low incentive to disrupt.

The paper is structured as follows: Section 2 outlines related work. Section 3 describes the system model attackers, and anonymity estimation. Section 4 discusses system parameters, game theory concepts and the modeling paradigm. Section 5 evaluates adversary, designer, and user strategies in DC-nets. Then we analyze anonymity metrics according to their underlying assumptions in Section 6. Finally, the paper is concluded in Section 7.

2 Related Work

The effectiveness of anonymity techniques is measures using anonymity metrics. Such measures refer to the probability that a powerful adversary becomes able to identify subjects of an anonymous communication. The measures make strong assumptions on available data, e.g. the a posteriori knowledge of an adversary [2] and involve the number of (honest and dishonest) users. Díaz, et. al examine probabilistic attacks, where an attacker weakens anonymity by concluding statements like 'with probability p, subject s is the sender of the message'. Their measure aggregates probabilistic information using information entropy [3]. Tóth, Hornák and Vajda stress the diverse anonymity levels of system participants. They define a metric of local anonymity, which refers to messages of a specific user. Therefore their prediction is more fine-grained than the anonymity computed in average for the whole system. Their paper shows the relevance of user-specific anonymity measures [4].

Other research studies the economic dimension of anonymity systems. Fulfilling security goals often relies on correct behavior of multiple parties, e.g. users. Dingledine and Mathewson review the influence of participants' habits on the anonymity received [5]. Acquisti, Dingledine, and Syverson explore privacy preferences in an anonymity system using game theory. E.g. volunteering as a mix node enhances the level of anonymity. If operating a mix is too costly, the anonymity system can be used as proxy. Cost has an impact on the participant behavior and, in consequence, on the size of the anonymity set. The duality between secure anonymity and communication reliability is introduced [6]. While they consider the impact of strategic users to the anonymity system, it is left open how design parameters facilitate sufficient anonymity to the users.

3 Modeling Dining Cryptographer Networks

The *Dining Cryptographers* (DC) protocol provides anonymity of the sender [7]. In each communication round, each participant either sends a message or contributes an empty frame. The coding scheme superimposes the message content with additional data. For this reason, the DC protocol establishes pairwise one-time pads (OTP). Each receiver of the broadcast decodes the superimposed message, that is assembled from all messages sent in this communication round. At that stage, the message sender is indistinguishable among the DC-net participants.

Attacker Models. Anonymity is threatened by several kinds of attack rendered against the DC protocol. We assume that any attacker prevents being identified as adversary, e.g. conceals its activity. A *global passive adversary* (GPA) is able to observe all communications. If all other participants of the anonymity set collude ($n-1$ attack), sender anonymity is broken. If an anonymous sender is linkable in multiple observations, the *intersection attack* narrows down the set of possible candidates. For this reason, the anonymity set must not be available to system participants.

Strategic User Behavior. In our experiments, a user decides strategically whether to join the DC-net or not. The cost of participation is additional traffic, both for contributing cover traffic broadcasts and for subscribing to the broadcast. If bandwidth usage had not been subject to economical considerations, a multitude of cover traffic sources would establish network anonymity on the Internet. Any user of an anonymity system considers the following questions: How much traffic will be received during participation in the DC-net? What level of sender anonymity will the user gain?

An adversary can also abuse the economical considerations of a user. The *disrupter attack* raises the cost of participation. If the message coding is not robust to collisions and a slot carries multiple messages at a time, message decoding fails. The disrupter attack provokes random collisions. This increases delay and requires retransmissions, which increases bandwidth consumption. Thus, an adversary can control cost of participation and cause users to leave the DC-net. This degrades sender anonymity of the remaining participants as the anonymity set size decreases.

Estimate Anonymity. The GPA is able to compute the anonymity set using the transitive closure using the relation of observed traffic. Anonymity metrics measure the knowledge of an adversary, e.g. the probability distribution within the candidate set of subjects. Without further knowledge, a uniform probability distribution is given: $p_i = (|\text{anonymity set}|)^{-1}$. This probability estimates the level of sender anonymity. Because of the lack of identification, distributed estimation of the anonymity set cannot be protected against malicious tampering. In general, users are not able to determine the anonymity currently received (cf. Section 6). The DC broadcast messages allow for anonymity estimation, because all participants contribute cover traffic. DC-nets hereby fulfill the necessary requirement for strategic behavior of users.

4 Strategic DC-Net Design

The anonymity of a Dining Cryptographers network results from the behavior of three players: designer, user, and adversary. An earlier study showed that varying cost together with individual anonymity preferences has an effect on the participant number [6] and, in consequence, also has an impact on the anonymity level. The players in the design game of an anonymity system have individual goals and therefore differing utility functions.

Design Dilemma. In our model, users aim for anonymity at a reasonable cost. According to the individual threshold, a user leaves the system if the cost of participation exceeds its benefit of sender anonymity. The designer wants to establish a high anonymity level for the system users. Sender anonymity is enhanced if the system attracts many users, i.e. has low cost of participation. Therefore, we expect that facilitating algorithms with low bandwidth usage result in a raise of participants. On the other hand, such coding schemes are vulnerable to attacks. Our study evaluates possible strategies, since no trivial solution exists. The objective of the adversary is to hinder anonymous communications or raise the cost of participation. An malicious participant can disrupt communications by generating collisions.

How should the designer set system parameters? The strategic behavior of all participants aims to enhance utility. *Nash equilibria* (NE) defines a set of game outcomes, where none of the players is able to improve its utility by changing its own strategy. Strategic behavior always results in a NE, if one exists. The utility functions correspond to anonymity set size (designer), sender anonymity at reasonable cost (user), and disrupting communications without being identified (adversary). Unfortunately, these optimization goals conflict with each other. In games containing such a *dilemma*, strategic behavior of all participants leads to utility loss. This is similar to the prisoner's dilemma, where lack of cooperation degrades utility of all participants. Earlier studies characterized utility functions that cause dilemmas in non-cooperative games [8]. The coding scheme used in the DC-net and anonymity preference of the user influence the utility functions and, possibly, neutralize the dilemma. Our study computes NE of *non-cooperative* games, where players know the utility functions of each other and consider strategies for maximum utility. Access to the utility functions is commonly assumed in game theoretic studies of cooperative systems, e.g. [9]. The analysis concludes parameter settings that avoid dilemma games. Then we relate the results to *sequential games*, where incomplete knowledge requires players to act defensively. Sequential games provide better models of anonymity systems because users and adversaries have perfect information about the design parameters.

Game theoretic modeling. Game theory studies the results of game strategies. Each player i chooses a strategy $s_i \in \Sigma_i$ from its strategy set, e.g. a user can either participate in the DC protocol or leave. A game with three players is defined as a tuple $G = ((\Sigma_i)_{i=1..3}, E, (u_i)_{i=1..3})$ with the strategy set Σ_i, the outcome set $E = \Sigma_1 \times \Sigma_2 \times \Sigma_3$, and the utility function $u_i := E \rightarrow \mathbb{R}$. Interaction between players influences the outcome $e \in E$ of a game. If NE are not unique, as in our study, the choice of strategy has no effect to the payoff.

In an iterated game, strategic behavior considers information about recent behavior of other players. The objective in iterated games is to maximize the average utility. This behavior is different to one-shot games, where strategies maximizes utility of a single round. The corresponding notion is the *mixed strategy* $x = (x_1, \ldots, x_n)$ with $\sum_{i=1..n} x_i = 1$ and n pure strategies. It executes each pure strategy s_i with a given probability $x_i \geq 0$.

Anonymity Design Parameters. The strategy sets Σ_i of designer, user, and adversary are listed in Table 1. A major design question is the choice of the algorithm, i.e. how messages are encoded for broadcast. The first parameter, α, defines collision robustness. Parameter β controls the ability to identify a disrupter. The third parameter, γ, defines the user anonymity preference. Chaum's XOR-based coding [7] consumes little bandwidth, but is highly vulnerable to collisions. For $\alpha = 0$ our simulation uses XOR coding. Between honest participants collisions can be avoided using slot reservation. After a collision, senders wait for a random time before they retry. Disrupter attacks can maliciously hinder any communication. Novel coding schemes have come up. Collisions in DC broadcasts can be recognized using bilinear maps [10]. The DC protocol becomes non-interactive with this coding ($\alpha = 1, \beta = 0$), as multiple senders can concurrently transmit messages. An alternative coding identifies disrupters, who repeatedly interfere with ongoing communications [11]. As an adversary hides from identification, the design parameters $\alpha = 1, \beta = 1$ counter disruptions from strategic adversaries. Another parameter models the anonymity preference of a

Table 1. Overview of players, their strategy sets and system parameters

Player	Property	Description
Designer	Objective	Provide high level of anonymity
	Σ_1	Efficient ($s_1 = 0$) vs. adversary-robust ($s_1 = 1$)
	α	Preference for defense rather than attractiveness to users
	β	Capability to identify malicious disrupters
User	Objective	Communicate anonymously with low cost
	Σ_2	Participate ($s_2 = 0$) vs. leave ($s_2 = 1$)
	γ	Demand for sender anonymity
Adversary	Objective	Disrupt DC communications, but remain unidentified
	Σ_3	Conforming ($s_3 = 0$) vs. disrupt ($s_3 = 1$)

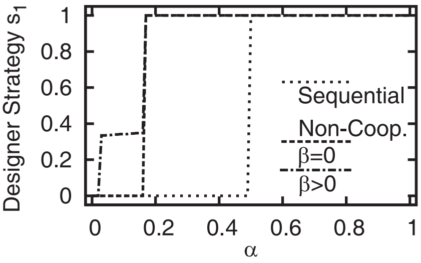


Fig. 1. Comparison of designer strategies by collision robustness with $\gamma = 1$

specific user, i.e. what effort is considered reasonable to sender anonymity (large effort $\gamma = 1$, no effort $\gamma = 0$).

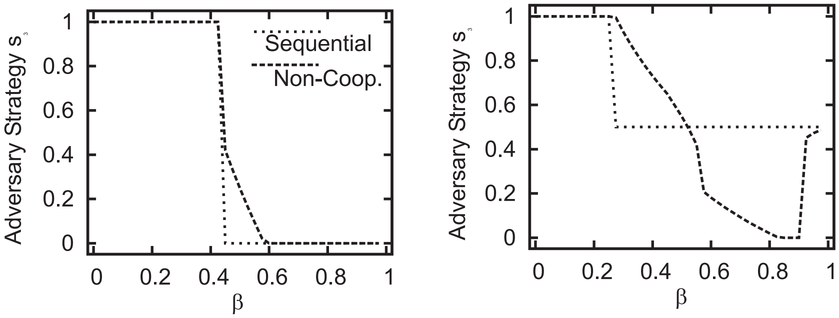
5 Evaluation

For the study of anonymity design parameters, utility functions are derived from the prisoner's dilemma ($T = 5, R = 3, P = 1, S = 0$) and involve the design parameters described in Table 1. Strategic players in non-cooperative games maximize their own utility, considering the behavior of other players. Our analysis is based on Nash Equilibria (NE) of the corresponding game, which define mixed strategies with maximum utility for each player. In sequential games, a defensive strategy minimizes utility loss (max-min algorithm). The system should be parameterized using α, β so that many users contribute to sender anonymity and strategically behaving adversaries do not disrupt anonymous communications.

Design Parameters. The first design choice is between cost-efficient anonymity or robustness against collisions. The study evaluates maximum utility strategies under a disrupter attack $s_3 = 1$. The designer's strategy s_1 resulting from the Nash Equilibria in non-cooperative games with ($\beta > 0$) and without ($\beta = 0$) disrupter identification and the sequential game strategy are shown in Figure 1.

The major trend is that sender anonymity benefits from attack countermeasures, unless there is a strong preference for low-bandwidth utilization. This preference together with disrupter identification is best answered by a 1 : 2 mixture of efficient coded rounds and collision-robust rounds. A mixed strategy NE is a typical result in dilemma games. When comparing a non-cooperative and a sequential game, the designer's strategy deviates for $0.2 \leq \alpha < 0.5$. This originates in the lack of feedback from user and adversary. In the sequential game, the designer benefits from low preference $0 < \alpha < 0.15$, which results in good anonymity, as many users join. In a non-cooperative game, the designer considers strategies of the other players. An adversary can impact communication efficiency by disrupting and wasting bandwidth. Therefore, in the non-cooperative game, an attacker-robust coding scheme is more beneficial.

The capability of disrupter identification indeed influences the adversary's strategy. The best adversary's strategy corresponding to disrupter identification



(a) $\alpha = \gamma = 0.5$

(b) $\alpha = 1, \gamma = 0$

Fig. 2. Impact of a disrupter-identifying algorithm β

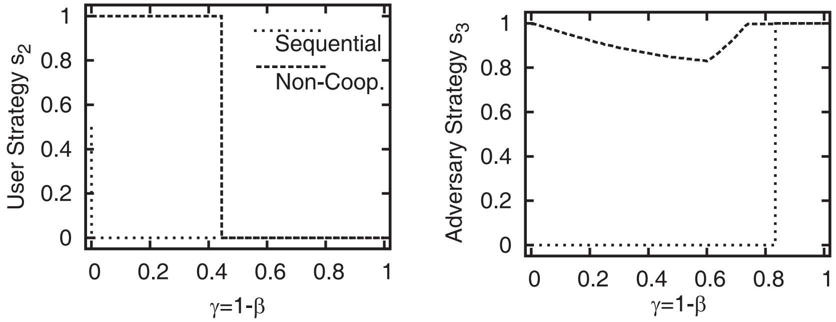
β is shown in Figure 2(a). Here, the designer balances robustness and efficiency $\alpha = 0.5$. If the adversary does not fear identification $\beta < 0.42$, it will disrupt communications. The strategy of the sequential game differs from the non-cooperative game NE for $0.42 < \beta < 0.6$. The adversary will omit disruption when it considers the designer's strategy, due to the enabled disrupter identification.

Does a collision-robust coding scheme make disrupter identification obsolete? Figure 2(b) shows the NE with varied disrupter identification. The adversary's strategy in a non-cooperative game adapts to a broad set of mixed strategies, 50% attacks for both $\beta = 0.5$ and $\beta = 1.0$ and no attacks for $\beta = 0.825$. This results from negotiation with the low-anonymity demand user strategies $\gamma = 0$. The underlying dilemma becomes also visible in the sequential game. The adversary achieves best utility by alternating disrupting and conforming behavior for $\beta > 0.3$. These results indicate that disrupter identification is necessary, as it influences the adversary to throttle its attack rate. The adversary exploits the countermeasure, which requires multiple collisions for correct disrupter identification.

Summarizing, control of the parameters α, β limits malicious disruptions. The overall design recommendations for equivalent utility functions are $\alpha > 0.5$ and $0.825 < \beta < 0.9$. Using this parameter set the designer can cut down the additional workload for the robust coding scheme. The mixed strategy of the adversary indicates the probability to randomly interfere with communications. The disrupter's defense against being identified is to attack from multiple DC-net participants. In this case, a low setting of β may fail to maintain communications.

Our results show how to resolve the game-theoretic dilemma. If the anonymity system is designed accordingly, strategic behavior increases utility (at least for designer and user). This facilitates that strategic behavior enhances a player's payoff. Then, strategic behaving participants have positive impact on the anonymity.

User Strategies for Anonymity. How do users behave if the design parameter mismatches their anonymity preference? This is the case for low-anonymity users with cost intensive disrupter identification $\beta = 0, \gamma = 1$ and users with high anonymity preference without disrupter identification $\beta = 1, \gamma = 0$. For the next



(a) User Strategy

(b) Adversary Strategy

Fig. 3. Impact of user anonymity demand γ and disrupter identification $1-\beta$ ($\alpha \geq 0.5$)

Table 2. Categories of anonymity measures

Category	Trust	Prediction	Example
Assured Anonymity	N	N	mix queue state with cover traffic
Policy-enforced Anonymity	Y	N	number-triggered pool mix
Reported Anonymity	N	Y	state externally aggregated
Perceived Anonymity	Y	Y	broadcast networks

experiment we choose $\beta := 1 - \gamma$ and examine resulting NE in Figure 3. User and adversary form a dilemma in the non-cooperative game for $\gamma = 0, \beta = 1$, where the user leaves the system and the adversary disrupts. The user participates for $\gamma > 0.45, \beta < 0.55$, but the adversary disrupts with high probability. In the sequential game, the user participates if a robust coding is used. Figure 3(a) displays the adversaries strategy in non-cooperative and sequential games. The adversary clearly benefits from negotiation of strategies, while pre-established strategies hinder attacks for $\gamma = 1 - \beta < 0.85$. This is an encouraging result, as the designer is able to mitigate attacks by choosing system parameters accordingly. Furthermore, the sequential game indicates that users benefit from participation, unless they do not value sender anonymity at all – $\gamma > 0$.

6 Requirements for Strategic User Behavior

A strategic user decision considers cost and benefit of participating in the anonymity system. While the cost is determined through design parameters, the anonymity level results from participation of other users. We propose four categories of anonymity measures shown in Table 2.

Perceived anonymity provides a prediction based on own experience in the past, i.e. the user has acquired knowledge directly. Externally provided information is not involved. DC-net participants receive all broadcasts and can compute the anonymity set of that round. Sender anonymity only fails if an adversary pools enough keys to reveal a DC message. Because a sender shares OTP keys with all other participants, changes to the anonymity set are actively distributed; otherwise the broadcasts cannot be decoded.

A superior choice is a *policy-enforced* anonymity mechanism, where the suggested level of anonymity is reached or messages will not be delivered. This requires trust into a third party, who has to enforce the policy. E.g. amount-triggered pool mixes queue incoming messages until a certain message count is reached. This weakens the adversary's knowledge if the mix is under observation.

Reported anonymity assumes trust into the reported anonymity set, which refers to a past system state. The anonymity prediction only holds if the system does not change many before the resulting strategy is executed. Reported anonymity is applied in large anonymity systems, where the anonymity set is expected to change only marginally over time. An example is the AN.ON system, which reports the number of participating users [12].

From an analytical viewpoint, only a non-predictive level of anonymity, whose evaluation does not rely on trusted third parties enables strategic reasoning. *Assured* anonymity determines the anonymity level of a message to be sent in the future. The evaluation must be independent from external influences. If cover traffic is contributed from multiple, non-colluding participants, a pool mix is able to determine assured anonymity. In peer-to-peer anonymity systems, each node acts as mixer [13]. The number of queue messages defines the lower bound of the anonymity set.

Concluding, the actual user behavior depends on design parameters, but also the user's ability to determine the anonymity level that the system provides. If an anonymity system participant cannot determine the anonymity level, it may prefer to leave the system. The ability to determine the anonymity level of future messages is suggested as future work.

7 Conclusions

If a user considers sender anonymity as a large advantage, he will accept the cost of participation. Dining Cryptographer networks rely on the willingness of many users in order to establish a good level of sender anonymity. Our work considers design parameters and analyzes the impact of participant strategies on the anonymity level. Parameters in the design of DC-nets contain dilemmas, where strategic behavior does not enhance anonymity. Our approach tunes system parameters to resolve strategic dilemmas and enhance anonymity. We classify measures that predict and guarantee a certain anonymity level and explicitly model trust relationships. Strategic behavior of participants must take these criteria into account in order to determine utility. In non-cooperative games, strategic players predict the behavior of other participants to evaluate their maximum benefit. In sequential games, the knowledge about adversaries is incomplete and their strategy cannot be predicted. Our simulation compares strategies of non-cooperative games with strategies in sequential games. We identify system parameters, which allow for dilemma-free games in DC-nets and, therefore, allow that strategic behavior enhances anonymity.

References

[1] Pfitzmann, A., Hansen, M.: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology (2008) http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[2] Serjantov, A., Newman, R.E.: On the anonymity of timed pool mixes. In: SEC – Workshop on Privacy and Anonymity Issues, vol. 250, pp. 427–434. Kluwer, Dordrecht (2003)

[3] Díaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 54–68. Springer, Heidelberg (2003)

[4] Tóth, G., Hornák, Z., Vajda, F.: Measuring Anonymity Revisited. In: Nordic Workshop on Secure IT Systems, pp. 85–90 (2004)

[5] Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: Workshop on the Economics of Information Security (2006)

[6] Acquisti, A., Dingledine, R., Syverson, P.: On the economics of anonymity. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 84–102. Springer, Heidelberg (2003)

[7] Chaum, D.: The dining cryptographers problem: unconditional sender and recipient untraceability. Journal of Cryptology 1, 65–75 (1988)

[8] Delahaye, J.P., Mathieu, P.: The iterated lift dilemma or how to establish metacooperation with your opponent. Chaos & Society (1996)

[9] Mahajan, R., Rodrig, M., Wetherall, D., Zahorjan, J.: Experiences applying game theory to system design. In: ACM SIGCOMM workshop on Practice and theory of incentives in networked systems (PINS), pp. 183–190. ACM, New York (2004)

[10] 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)

[11] Bos, J.N., den Boer, B.: Detection of Disrupters in the DC Protocol. In: Workshop on the theory and application of cryptographic techniques on Advances in cryptology, pp. 320–327 (1989)

[12] Berthold, O., Federrath, H., Köpsell, S.: Web MIXes: A system for anonymous and unobservable Internet access. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 115–129. Springer, Heidelberg (2001)

[13] Rennhard, M., Plattner, B.: Introducing MorphMix: peer-to-peer based anonymous internet usage with collusion detection. In: ACM workshop on Privacy in the Electronic Society (WPES), pp. 91–102 (2002)

Obligations: Building a Bridge between Personal and Enterprise Privacy in Pervasive Computing

Susana Alcalde Bagüés[1,2], Jelena Mitic[1], Andreas Zeidler[1],
Marta Tejada[2], Ignacio R. Matias[2], and Carlos Fernandez Valdivielso[2]

[1] Siemens AG, Corporate Technology
Munich, Germany
{susana.alcalde.ext,jelena.mitic,a.zeidler}@siemens.com
[2] Public University of Navarra
Department of Electrical and Electronic Engineering
Navarra, Spain
{tejada.43281,carlos.fernandez,natxo}@unavarra.es

Abstract. In this paper we present a novel architecture for extending the traditional notion of access control to privacy-related data toward a holistic privacy management system. The key elements used are obligations. They constitute a means for controlling the use of private data even after the data was disclosed to some third-party. Today's laws mostly are regulating the conduct of business between an individual and some enterprise. They mainly focus on long-lived and static relationships between a user and a service provider. However, due to the dynamic nature of pervasive computing environments, rather more sophisticated mechanisms than a simple offer/accept-based privacy negotiation are required. Thus, we introduce a privacy architecture which allows a user not only to negotiate the level of privacy needed in a rather automated way but also to track and monitor the whole life-cycle of data once it has been disclosed.

1 Introduction

Over the last few years *privacy topics* have attracted the attention of many researches working in the field of *Pervasive Computing*. The existing common understanding is: the envisioned age of invisible computing is only feasible if people have control over the circumstances under which their personal data is disclosed and how it is processed thereafter. The demand is clear: we should design pervasive computing environments aware of their users' privacy preferences. So far, most efforts are centered around privacy control for enterprises, like E-P3P [1] and EPAL [2]. However, we argue that pervasive computing settings demand an additional level of *personal privacy* complementing enterprise privacy in important aspects. Personal privacy is concerned with maintaining a user's privacy preferences. In our opinion, for guaranteeing an individual's right for privacy, it is necessary to empower a user to decide on the exchange of personal data on a much finer-grained level than possible today. Apart from such mechanisms that provide access control for commercial use, and more recently obligations management [3], users should have their own personalized *context-aware privacy access control*, and additionally the possibility of monitoring the *post-disclosure life-cycle*

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 173–184, 2008.

of the data transmitted. The goal is to enable users to monitor the access, use and deletion of data, also *after* the data was disclosed. Today, this is only possible to the extent that an enterprise "promises" to respect a user's privacy preferences.

We enable post-disclosure monitoring by introducing *obligations* as an independent entity within our User-centric Privacy Framework (UCPF) [4]. *Wikipedia* defines an obligation as *"a requirement to take some course of action"*. In our work presented here, we leverage this notion of an obligation as a required description of regulation on the processing of personal data when being disclosed to third-parties. In this paper, we describe how we add specialized layers for privacy management in order to realize a holistic privacy control, able to fulfill a user's privacy preferences. The key idea is to combine personal and enterprise privacy in an appropriate way. For us it is clear that personal privacy demands differ substantially from those assumed by enterprises, since personal privacy is a much more intimate concern than an enterprise's requirement to meet existing legislations.

This paper is structured as follows: Section 2 compares the requirements for personal privacy protection with those for enterprises. Section 3 then introduces our own privacy framework. Section 4 and 5 are dedicated to our approach for a holistic privacy management based on obligations. The following Sections 6 and 7 are summarizing related work and conclude this paper also indicating directions for future work.

2 Personal and Enterprise Privacy in Pervasive Computing

Pervasive computing scenarios entail the deployment of a large number of *Context-aware Mobile Services* (CAMS) and along with them a "pervasive sensing" of context information related to a person at any time and any place. Therefore, individuals will require automatic mechanisms to control when context is revealed without the need to set their privacy preferences each time and for each service separately. Even the large number of services alone will make a manual per-use authorization of access to personal data (as required by law) an impossible task. Furthermore, individuals will want mechanisms to monitor that enterprises use disclosed data only for fulfilling the requested purpose and nothing else. The challenge here is to meet the individual's expected level of privacy while at the same time dynamic information is revealed in mobile, inherently changing and distributed settings.

Today, enterprises and organizations offer mainly privacy protection mechanisms oriented toward the long-term use of services. They consume personal data, which is classified as *static* in [5], e.g. account number or address. In contrast to the dynamic and short-lived relations typically found in pervasive and mobile settings, where data usually is provided and used only once in a single request. In the latter setting, obviously it is no longer possible to spend the time and effort to define or acknowledge privacy preferences at the moment of use, which is normal for Internet services or company applications. An "offline" solution is needed where a user can define the privacy policy with which a newly discovered service is used; beforehand of actually being in the situation of using it. Our proposal is to add specialized layers of privacy management to give a user a maximum control over the specification and enforcement of his privacy.

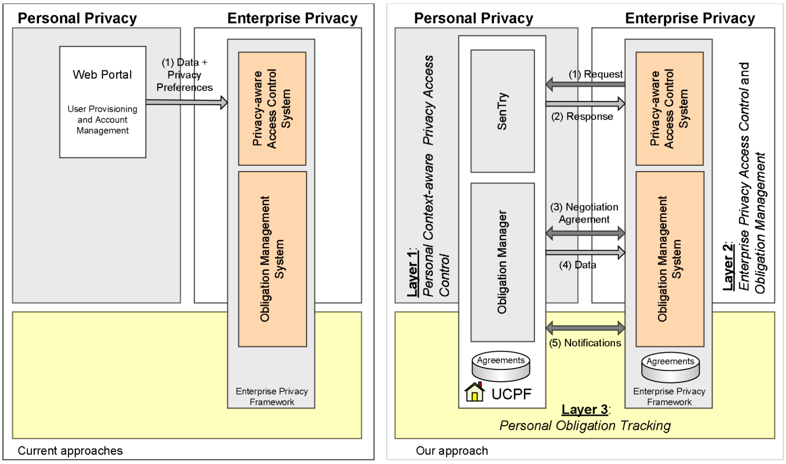

Personal Privacy
Enterprise Privacy
Web Portal
User Provisioning and Account Management
(1) Data + Privacy Preferences
Privacy-aware Access Control System
Obligation Management System
Enterprise Privacy Framework
Current approaches
Personal Privacy
Enterprise Privacy
Layer 1: Personal Context-aware Privacy Access Control
SenTry
Obligation Manager
Agreements
UCPF
(1) Request
(2) Response
(3) Negotiation Agreement
(4) Data
(5) Notifications
Privacy-aware Access Control System
Obligation Management System
Agreements
Enterprise Privacy Framework
Layer 2: Enterprise Privacy Access Control and Obligation Management
Layer 3: Personal Obligation Tracking
Our approach

Fig. 1. Privacy Architectures

Figure 1 compares two different situations, on the left hand side the current use of obligations by enterprises, situation applicable to the long-lived service bindings. And on the right hand side our approach of a holistic privacy control scheme integrating access control and obligation tracking on the user side.

In order to address named challenges for privacy in pervasive computing, we have developed a novel privacy architecture consisting of three layers, namely: 1) Personal context-aware privacy access control, 2) Enterprise access control and obligation management, and 3) Personal obligation tracking. These layers are complementary and depend on each other to guarantee a holistic privacy protection. For instance, personal privacy alone cannot protect data once it was transmitted and must rely on enterprise privacy mechanisms. For the rest of the paper we assume that a privacy protection middleware on the enterprise side is in place, capable of handling and enforcing obligations.

The first privacy layer is provided by our privacy framework called UCPF (cf. Section 3). It acts as a personal access filter to delimit when, what, how and who gets permission to access a user's personal data. In pervasive computing scenarios mostly this is related to disclosing context information, e.g. location or activity. Obviously, for users it is desirable to automate such frequent decisions as much as possible and also to have their current context taken into account. For instance, in the example: "Bob allows the disclosure of his location to his employer when his activity state is *working*", the activity of Bob must be checked to decide whether his location is disclosed or not. Therefore, the UCPF's policy system (SenTry) has as design requirement to be *context-aware* [6], in the sense that the evaluation process of a policy might involve consulting a user's context or peer context (e.g. requester) against the applicable policy constraints. We argue that leaving the enforcement of such constraints to a third-party (e.g. enterprise access control system) would not be advisable since it entails the disclosure of sensitive information during the policy evaluation (Bob's activity). Nevertheless, this privacy layer can only address situations where information is disclosed for the present use. But it does not cover cases where information may be stored for future use in a potentially different context. So, the user has to trust the service to adhere to the legal regulations for enterprises. Here is where the second and third privacy layers are introduced to make users aware of the whole life-cycle of information once it was disclosed.

The second privacy layer is the enterprise privacy access control and obligation management depicted in Figure 1, right hand side. Once data was transmitted, after following the evaluation and enforcement process of the appropriate user's privacy policy in the UCPF, the enterprise service takes over the task of protecting the data. Enterprises are obliged by law to control all accesses to the data gathered from their users. In order to comply with current legislation, enterprise guidelines [7] and individual privacy preferences, enterprise service providers not only should apply traditional access control but also actively accept and enforce privacy obligations from the users. This notion of privacy enforcement is in accordance with the work of Hewlett Packard [3] as part of the European Union project PRIME [8]. Obligations impose conditions for the future that the enterprise is bound to fulfill [9], e.g "My data must be deleted within one month" or "Send a notification when Bob requests Alice location more than N times". This is of vital importance since an enterprise is the only entity in an interaction chain, see Figure 2, able to deal with future situations.

The idea of an enterprise privacy middleware able to enforce obligations based on a user's privacy preferences has been inspired by the work of HP in its Obligation Management System (OMS) [10]. In the OMS framework users can explicitly define their privacy preferences at disclosure time or at any subsequent point of time, e.g., through a web portal. Such privacy preferences are automatically translated into privacy obligations based on a predefined set of templates. As mentioned before this approach is valid for services with long-lived bindings but due to the dynamic nature of CAMS a different solution is needed to automate the exchange of data and privacy preferences. To do so, we impose privacy on enterprises by employing an automatic negotiation protocol over a set of obligations, which contain a user's privacy preferences related with the service used.

The third privacy layer realizes the requirement to empower users of being aware of the "life-cycle" of data after being transmitted. Here, we have developed a set of strategies to reach a trust relationship based on a notification protocol on the agreements stored, at the time of the disclosure between a UCPF and some enterprise service. Agreements include the set of obligations defined by a user in his privacy policy, more details can be found in Section 4.

3 User-Centric Privacy Framework

A first prototype of the UCPF [4] has been developed to be tested on the residential gateway for the Siemens Smart Home Lab. The residential gateway provides access to home-based services from inside and outside the home environment. The incorporation of the UCPF adds privacy control and context brokering as separate functionalities and lets inhabitants interact with outside CAMS. Part of the implementation of the UCPF was incorporated into the privacy framework of the IST project CONNECT [11] as well.

As shown in Figure 2, the UCPF consists of six main functional elements, the SenTry or policy system, the Obligation Manager (OM), the Sentry Registry, the Context

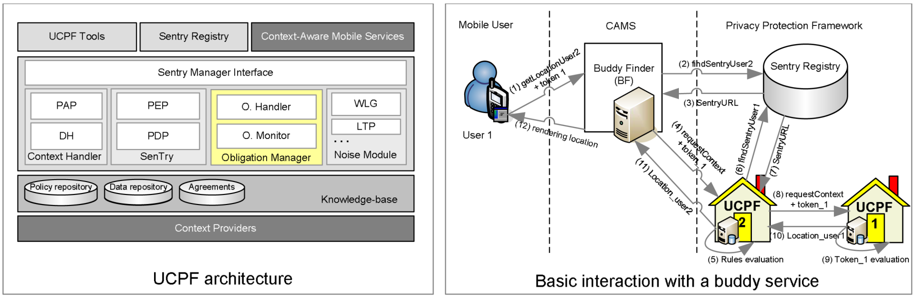


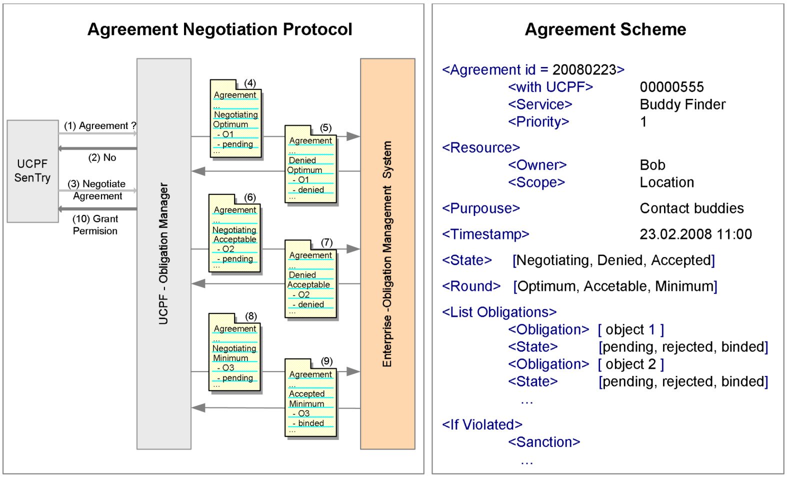
Fig. 2. UCPF overview

Handler (CH), the SenTry Manager Interface (SMI), and the Noise Module (NM). The SenTry is the major architecture building block. It was developed in JAVA on top of the Java Expert System Shell, called Jess. A SenTry instance manages the context disclosure of a user to third parties, based on a set of personal privacy policies defined with the SeT policy language [6]. We benchmarked the performance of the SenTry together with the policy language by using a repository of 500 rules grouped into 6 policies. In average a request took less than 50ms to be evaluated on a standard PC, which seems to be a reasonable performance for the application scenarios considered.

The Obligation Manager, see next Section, negotiates agreements and tracks the obligations agreed on by third parties. The SenTry Registry is the only component that is not co-located with the rest of the elements on the gateway. This component is shared among sentries instances and located in the Internet. It tracks the availability of people's context and provides the pointer to the appropriate SenTry service instance, see Figure 2 right hand side. The interface between SenTry and Service Registry is facilitated by the Context Handler (CH). It supports the identification of external sources of context e.g. for the evaluation of *Foreign Constraints* [12]. Furthermore, the CH acts as a mediator between SenTry and externals context providers. The interaction of end-users with the UCPF is made possible through the Sentry Manager Interface (SMI). It is implemented as an API used to generate, upgrade or delete privacy policies, receive information about the current applicable policies, or getting feedback on potential privacy risks and obligations state. The Noise Module (NM) is a modular component that incorporates additional tools to the policy matching mechanism, e.g. obfuscation and *white lies* [13].

4 Building a Bridge toward Holistic Privacy

In Section 2 the idea of a holistic privacy protection was introduced together with its dependency on the collaboration between a personal privacy framework (the UCPF) and an enterprise privacy system. The question we address now is: *How can this collaboration be established?*, The main problem obviously is that users still have to trust to some degree in enterprises' "promises". Obligations are used to create automatic bindings between both parts, and ensure that data protection requirements are adhered to. However, in cases where those bindings cannot be monitored, checking the compliance


Agreement Negotiation Protocol
UCPF SenTry
(1) Agreement ?
(2) No
(3) Negotiate Agreement
(10) Grant Permision
UCPF - Obligation Manager
Enterprise -Obligation Management System
(4) Agreement ... Negotiating Optimum - O1 - pending ...
(5) Agreement ... Denied Optimum - O1 - denied ...
(6) Agreement ... Negotiating Acceptable - O2 - pending ...
(7) Agreement ... Denied Acceptable - O2 - denied ...
(8) Agreement ... Negotiating Minimum - O3 - pending ...
(9) Agreement ... Accepted Minimum - O3 - binded ...
Agreement Scheme

```
<Agreement id = 20080223>
        <with UCPF>       00000555
        <Service>         Buddy Finder
        <Priority>        1
<Resource>
        <Owner>           Bob
        <Scope>           Location
<Purpouse>                Contact buddies
<Timestamp>               23.02.2008 11:00
<State>   [Negotiating, Denied, Accepted]
<Round>   [Optimum, Accetable, Minimum]
<List Obligations>
        <Obligation>  [ object 1 ]
        <State>       [pending, rejected, binded]
        <Obligation>  [ object 2 ]
        <State>       [pending, rejected, binded]
          ...
<If Violated>
        <Sanction>
          ...
```

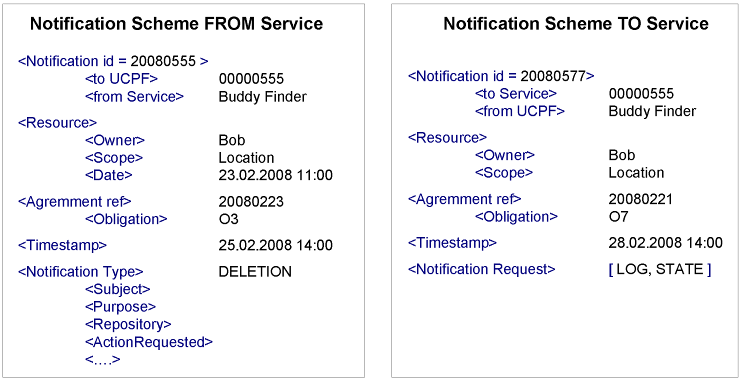
Fig. 3. Agreement Negotiation

with the obligation is almost impossible. The concept of *Non-observable Obligations* is described in the work of Hilty et al. [9], they suggest that a possible solution is the use of nontechnical means, such as audits or legal means. We propose instead the idea of employing *observable* bindings between personal and enterprise frameworks. This is realized by introducing an agreement negotiation protocol together with a trusted notification mechanism, both detailed below.

The agreement negotiation protocol, cf. Fig. 3, starts after the evaluation of a service request within a SenTry instance. If the *rule effect* compiled contains obligations, the *Policy Enforcement Point* (PEP) queries the OM for an agreement over the *pending obligations*, step 3 in Figure 3, and the OM launches the negotiation, steps 4 to 9.

This protocol enables per-service and per-user resource agreements negotiations that are guaranteed to terminate after at most three negotiation rounds. The Obligation Manager makes a first proposal in the *"Negotiating Optimum"* stage. The enterprise side cannot make any counter-proposal at this stage, since the user should not be involved during the negotiation. Therefore, it is limited to check the list of obligations attached and to reject or bind them. If the agreement is denied by the enterprise, which means that one or more obligations are rejected, the OM issues the second proposal: *"Negotiating Acceptable"* stage. It includes a new set of obligations where the rejected obligations of the first set are replaced by their *acceptable* equivalents. The enterprise service may accept the second proposal, or start the third and last round: *"Negotiating Minimum"* stage, in which a new set of obligations classified as *minimum* replaces those rejected. The goals of this negotiation strategy are: i) to allow more than "take or leave" situations, ii) to enable an automatic setup of user's privacy preferences, and iii) to execute the obligation binding process transparent to the user.

In a situation where an enterprise does not accept the third and last proposal, no agreement is reached and the priority of the rejected agreement is taken into account by the OM. Each agreement is labeled with a priority value, *one, two* or *three*. Priority one means that the service (enterprise) MUST accept the agreement otherwise permission



Notification Scheme FROM Service

```
<Notification id = 20080555 >
          <to UCPF>            00000555
          <from Service>       Buddy Finder
<Resource>
          <Owner>              Bob
          <Scope>              Location
          <Date>               23.02.2008 11:00
<Agrgemment ref>               20080223
          <Obligation>         O3
<Timestamp>                    25.02.2008 14:00
<Notification Type>            DELETION
          <Subject>
          <Purpose>
          <Repository>
          <ActionRequested>
          <….>
```

Notification Scheme TO Service

```
<Notification id = 20080577>
          <to Service>         00000555
          <from UCPF>          Buddy Finder
<Resource>
          <Owner>              Bob
          <Scope>              Location
<Agrgemment ref>               20080221
          <Obligation>         O7
<Timestamp>                    28.02.2008 14:00
<Notification Request>         [ LOG, STATE ]
```

Fig. 4. Notification Schemes

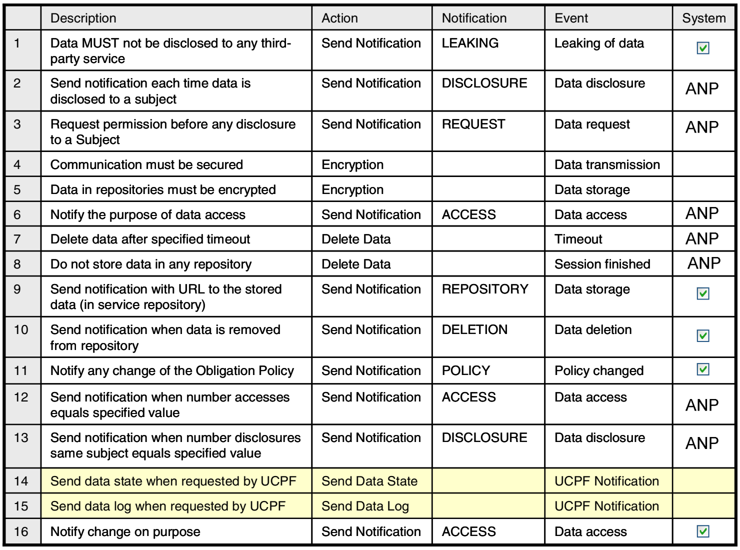
will be denied (step 10). Priority two means that the service SHOULD accept the agreement otherwise data quality will decrease in accuracy. A priority of three means that the service MIGHT accept the agreement and entails the disclosure of the requested data anyway but the user will be notified that no agreement was reached. A user may modify his privacy settings based on the obligations rejected.

Our approach to establish a trusted relationship between an enterprise service and the UCPF is based on the possibility to subscribe to notifications about the use of disclosed data. We introduce two complementary notification types as shown in Fig. 4. The notification template shown on the left hand side is used for notifying the UCPF (as subscriber) about the fulfillment or violation of an obligation by the service provider. We have defined seven notifications types, namely: DELETION, ACCESS, LEAKING, REPOSITORY, REQUEST, DISCLOSURE and POLICY. Depending on the notification type used, further parameters need to be provided. E.g. a DELETION notification does not have any parameter, on the other hand, a DISCLOSURE notification should include at least the *Subject* or *Service*, recipient of the data, and the *Purpose* of such disclosure. The use of notifications allows for monitoring the status of the active obligations and to define actions (penalties) in case of a violation. We introduced the tag *if Violated* (cf. Fig. 3) for this case. It describes the sanctions to be carried out once the OM observes such a violation.

The template on the right hand side of Fig. 4 is the notification scheme used by the UCPF to request a report on the state of or a list of the operations on a particular resource. In summary, notifications are leveraged for: i) enabling monitoring of active obligations, ii) auditing the enterprise service, iii) getting access to personal data in the service's repository (with REPOSITORY notification), iv) knowing when the service's obligation policy changes in order to re-negotiate agreements, and v) controlling when an active obligation is violated.

4.1 Model of Privacy Obligations in the UCPF

In collaboration with the IST project CONNECT, we created a set of 16 obligations as shown in Figure 5. They represent the privacy constraints that a user may impose on

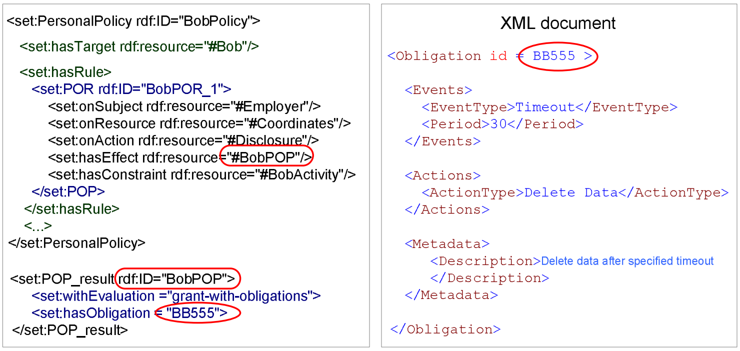


	Description	Action	Notification	Event	System
1	Data MUST not be disclosed to any third-party service	Send Notification	LEAKING	Leaking of data	☑
2	Send notification each time data is disclosed to a subject	Send Notification	DISCLOSURE	Data disclosure	ANP
3	Request permission before any disclosure to a Subject	Send Notification	REQUEST	Data request	ANP
4	Communication must be secured	Encryption		Data transmission	
5	Data in repositories must be encrypted	Encryption		Data storage	
6	Notify the purpose of data access	Send Notification	ACCESS	Data access	ANP
7	Delete data after specified timeout	Delete Data		Timeout	ANP
8	Do not store data in any repository	Delete Data		Session finished	ANP
9	Send notification with URL to the stored data (in service repository)	Send Notification	REPOSITORY	Data storage	☑
10	Send notification when data is removed from repository	Send Notification	DELETION	Data deletion	☑
11	Notify any change of the Obligation Policy	Send Notification	POLICY	Policy changed	☑
12	Send notification when number accesses equals specified value	Send Notification	ACCESS	Data access	ANP
13	Send notification when number disclosures same subject equals specified value	Send Notification	DISCLOSURE	Data disclosure	ANP
14	Send data state when requested by UCPF	Send Data State		UCPF Notification	
15	Send data log when requested by UCPF	Send Data Log		UCPF Notification	
16	Notify change on purpose	Send Notification	ACCESS	Data access	☑

Fig. 5. Obligations defined within the UCPF

an enterprise service when data is disclosed. In our definition, an obligation has two aspects; First, it is a second-class entity subject to the enforcement of a rule by the SenTry component and embedded as part of the rule effect (see Fig. 6, tag *hasEffect*) and to be compiled during the evaluation of a service request. And second, when an evaluation reaches the PEP and it contains obligations, it activates the agreement negotiation protocol as described on Fig. 3. Then, obligations are first-class entities used to convey personal privacy preferences.

In the representation of obligations basically we follow the scheme adopted by the HP framework to facilitate collaboration with the enterprise privacy system. Thus, obligations are XML documents with *Event*, *Action* and *Metadata* elements. Some tags



```
<set:PersonalPolicy rdf:ID="BobPolicy">
  <set:hasTarget rdf:resource="#Bob"/>
  <set:hasRule>
    <set:POR rdf:ID="BobPOR_1">
       <set:onSubject rdf:resource="#Employer"/>
       <set:onResource rdf:resource="#Coordinates"/>
       <set:onAction rdf:resource="#Disclosure"/>
       <set:hasEffect rdf:resource="#BobPOP"/>
       <set:hasConstraint rdf:resource="#BobActivity"/>
    </set:POP>
   </set:hasRule>
   <...>
</set:PersonalPolicy>

<set:POP_result rdf:ID="BobPOP">
    <set:withEvaluation ="grant-with-obligations">
    <set:hasObligation = "BB555">
</set:POP_result>
```

XML document

```
<Obligation id = BB555 >

  <Events>
    <EventType>Timeout</EventType>
    <Period>30</Period>
  </Events>

  <Actions>
    <ActionType>Delete Data</ActionType>
  </Actions>

  <Metadata>
     <Description>Delete data after specified timeout
     </Description>
  </Metadata>

</Obligation>
```

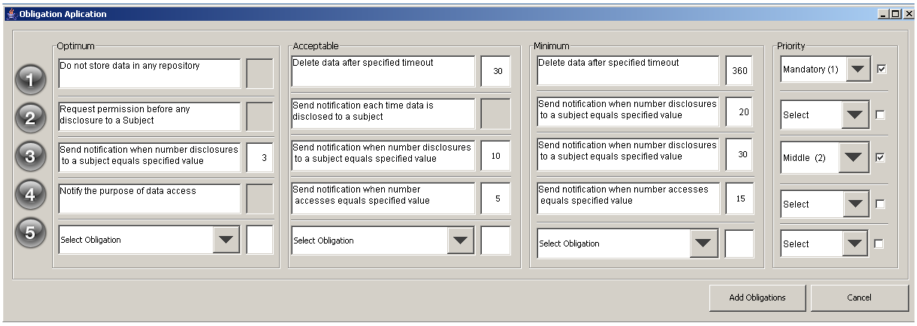
Fig. 6. Obligation's Example

were left out in our definition (e.g. Target), which only can be used together with HP's OMS. In Figure 6 a simple rule example is depicted to show how an XML obligation instance is created to be included in the agreement negotiation protocol (ANP). In this example Bob is allowing his employer to access his location based on the activity, but to delete his coordinates latest after 30 days. The rule is specified using our policy language SeT [6]. The instance of the rule effect (BobPOP) specifies that the result evaluates to "grant" but only if the service agrees on the obligation with id "BB555". Fig. 6, right hand side, shows the XML document referred by the named rule effect.

The table in Fig. 5 shows five special obligations marked as system. Those are mandatory obligations (deducted from current legislation), which are by default established independently of a user's preferences and beforehand of any commercial transaction with a service. The rest marked *ANP* mean that user might include them as a result of the evaluation of a rule and that they will be subject of the negotiation protocol. There are two more obligations highlighted that are obligations that allow the UCPF to audit the enterprise service.

5 Obligation Management from the User Perspective

Due to space restrictions we cannot really go into the details of obligation management. But still we want to give a short introduction to our ongoing work in this area. The question remaining at this point obviously is: *How can a user setup his obligation policies regarding optimal, acceptable, and minimum agreement?*. Figure 7 shows a screenshot of our current application prototype for managing sets of obligations. A user can specify a new rule for being added to his privacy policy and subsequently can allocate a set of obligations to it. A set always consists of the three mandatory obligation types "Optimum", "Acceptable", and "Minimum". These can be predefined and be re-used, obviously, and do not have to be defined separately each time. Implicitly they are always indirectly referenced by id and not by name or privacy rule. For example, in Fig. 6 the id of the obligation chosen is "BB555" which, for the sake of simplicity, is only a single obligation. In our real application the same id would refer to a set of three obligations corresponding to the three mandatory categories.



	Optimum		Acceptable		Minimum		Priority	
1	Do not store data in any repository		Delete data after specified timeout	30	Delete data after specified timeout	360	Mandatory (1)	☑
2	Request permission before any disclosure to a Subject		Send notification each time data is disclosed to a subject		Send notification when number disclosures to a subject equals specified value	20	Select	☐
3	Send notification when number disclosures to a subject equals specified value	3	Send notification when number disclosures to a subject equals specified value	10	Send notification when number disclosures to a subject equals specified value	30	Middle (2)	☑
4	Notify the purpose of data access		Send notification when number accesses equals specified value	5	Send notification when number accesses equals specified value	15	Select	☐
5	Select Obligation		Select Obligation		Select Obligation		Select	☐

Fig. 7. GUI prototype

In later implementations we hope to use the experiences gathered in field trials to improve the management for the use within different user groups. However, for the time being this has to be considered future work.

6 Related Work

The use of obligations in computer systems by itself is not a new topic. It has been largely used to specify actions that must be performed by some entity. In daily situations where people interact, individuals are held responsible for their own actions; they may be punished if they fail to do what they have promised. In 1996 Van the Riet et al. translated this concept to the "cyberspace". In [14] they conclude that although software entities cannot take real responsibility, their specification still should take into account what such entities must do and what happens if they do not fulfill what has been specified. For instance, within Ponder [15] obligations are event-triggered policies that carry-out management tasks on a set of target objects or on the subject itself, e.g. when a print error event occurs a policy management agent will notify all operators of the error and log the event internally based on an obligation.

In traditional access control systems obligations are coupled tightly to access control policies. An obligation is considered as an action that shall be performed by the system's Policy Enforcement Point (PEP) together with the enforcement of an access control decision. This is the approach followed by EPAL [2], where obligations are entities subordinated to access control. Similarly to EPAL, XACML [16] specifies the syntax and format of access control policies and related obligations. Within this popular access control language obligations are a set of operations associated with an XACML policy that must be executed by the PEP together with an authorization decision. In the work of Park and Sandhu [17] obligations are requirements that have to be fulfilled by a subject at the time of the request for allowing access to a resource. E.g., a user must give his name and email address to download a company's white paper. However, the mentioned approaches do not cover the main requirement of obligations in the context of post-disclosure life-cycle control. Here, obligations should be a binding agreement between two parts, the requesting service and the user, specifying actions to be carried out by the service's PEP after getting the data, at some point in the future. The main goal of these types of privacy obligations is not to constrain the access to the user data but to inform a remote requester of a user's privacy preferences, which should involve an agreement and its posterior tracking.

In Rei [18], an obligation describes an action that must be performed on an object by a distributed entity. An example of an obligation in Rei is "All members of a team are obliged to send weekly reports to the team leader". Rei uses obligations in a similar way to our work and introduces some common and important aspects, such as promises for the future and sanctions in case of violation. However, the Rei framework does not provide an enforcement model. Rei assumes that obligation management is done outside the policy engine although it is not clear how obligations are agreed upon or denied by a third party.

The work presented in [19] describes an approach to archive digital signed commitments on obligations between distributed parties. They introduced the *Obligation* *of Trust* (OoT) protocol, which executes two consecutive steps: *Notification of Obligation* and *Signed Acceptance of Obligation*. The OoT is built upon the XACML standard following its Web Services Profile (WS-XACML). The disadvantages of this approach are that it does not cater for the enforcement and monitoring of such obligations, on the one hand, and that it seems to be rather complicated for a common user to manage obligations following this protocol, on the other hand.

We propose a novel privacy architecture in which obligations can be managed by common users. Our framework provides privacy-aware access control, an agreement negotiation protocol over a set of obligations and its posterior tracking. In order to avoid the misuse of private data, once it was disclosed, we rely on the idea of an enterprise privacy middleware able to enforce obligations remotely. This notion of privacy obligations enforcement is in accordance with the work of Hewlett Packard [3] within PRIME [8], as already mentioned in Section 2.

7 Conclusions and Outlook

In this paper we present a novel architecture that extends privacy control in a substantial matter toward holistic privacy management. We introduced the notion of a binding obligation for each privacy-related resource. Such an obligation has to be accepted by a service whenever it requests access to private data. Obligations describe the rights and requirements for processing, storing or deleting data by the service. To avoid situations where obligations are not acceptable by a service and would lead to simple denial of service, we also defined a well-defined negotiation protocol for trying to find an agreement based on different classes of information- or service provided. However, we considered even this to be not far-reaching enough and introduced a third layer of privacy-related functionality: *the personal obligation tracking*, enabling post-disclosure life-cycle awareness. Obligations additionally can describe which information the client (user) wants to receive in order to track the usage of disclosed data after releasing it. We showed that this is an easy but effective way to enable trust between the client and the service. On the other hand, we are aware that we have to gather more experience with these mechanisms. Therefore, we currently are improving the client applications which allow users to maintain and manage their privacy-related settings. This is partly done in the context of the CONNECT project which serves as 'testbed' for the concepts and also provides input for realistic scenarios from different domains.

References

1. Karjoth, G., Schunter, M., Waidner, M.: The platform for enterprise privacy practices - privacy enabled management of customer data. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482. Springer, Heidelberg (2003)
2. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise Privacy Authorization Language (EPAL 1.2) Specification (November 2003), http://www.zurich.ibm.com/security/enterprise-privacy/epal/
3. Casassa Mont, M., Thyne, R.: A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 118–134. Springer, Heidelberg (2006)

4. Alcalde Bagüés, S., Zeidler, A., Fernandez Valdivielso, C., Matias, I.R.: Sentry@home - leveraging the smart home for privacy in pervasive computing. International Journal of Smart Home 1(2) (2007)
5. Price, B.A., Adam, K., Nuseibeh, B.: Keeping ubiquitous computing to yourself: a practical model for user control of privacy. International Journal of Human-Computer Studies 63, 228–253 (2005)
6. Alcalde Bagüés, S., Zeidler, A., Fernandez Valdivielso, C., Matias, I.R.: Towards personal privacy control. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 886–895. Springer, Heidelberg (2007)
7. Federal Trade Commission (FTC). Fair information practice principles. Privacy online: A (June 1998)
8. Camenisch, J., et al.: Privacy and Identity Management for Everyone. In: Proceedings of the ACM DIM (2005)
9. Hiltya, M., Basin, D.A., Pretschner, A.: On Obligations. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 98–117. Springer, Heidelberg (2005)
10. Casassa Mont, M.: A System to Handle Privacy Obligations in Enterprises. Thesis (2005)
11. The CONNECT Project, http://www.ist-connect.eu/
12. Alcalde Bagüés, S., Zeidler, A., Fernandez Valdivielso, C., Matias, I.R.: A user-centric privacy framework for pervasive environments. In: OTM Workshops (2), pp. 1347–1356 (2006)
13. Alcalde Bagüés, S., Zeidler, A., Fernandez Valdivielso, C., Matias, I.R.: Disappearing for a while - using white lies in pervasive computing. In: Proceedings of the 2007 ACM workshop on Privacy in electronic society (WPES 2007) (2007)
14. van de Riet, R.P., Burg, J.F.M.: Linguistic tools for modelling alter egos in cyberspace: Who is responsible? Journal of Universal Computer Science 2(9), 623–636 (1996)
15. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: Ponder: A language for specifying security and management policies for distributed systems (2000)
16. OASIS standard. eXtensible Access Control Markup Language. Version 2 (February 2005)
17. Park, J., Sandhu, R.: The uconabc usage control model. ACM Trans. Inf. Syst. Secur. 7(1), 128–174 (2004)
18. Kagal, L.: A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments. Phd Thesis, University of Maryland Baltimore County (September 2004)
19. Mbanaso, U.M., Cooper, G.S., Chadwick, D.W., Anderson, A.: Obligations for privacy and confidentiality in distributed transactions. In: EUC Workshops, pp. 69–81 (2007)

A User-Centric Protocol for Conditional Anonymity Revocation

Suriadi Suriadi, Ernest Foo, and Jason Smith

Information Security Institute, Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia
s.suriadi@isi.qut.edu.au, e.foo@qut.edu.au, j.smith@isi.qut.edu.au

Abstract. This paper presents and evaluates an improved anonymity revocation protocol. This protocol can be used to strengthen anonymity revocation capability in a privacy-enhancing identity management system. This protocol is user-centric, abuse-resistant, and it provides enforceable conditions fulfillment. We assume the existence of 1 honest referee out of t designated referees ($t > 1$) chosen by users, and no collusion between users and referees. The security and performance of this protocol are evaluated.

Keywords: privacy, user-centric identity, anonymity revocation.

1 Introduction

Anonymity revocation capability can be the weakest link in a privacy-enhanced environment, such as in a medical field. Failure to safeguard such a capability will render the overall privacy protection useless. Existing privacy-enhancing cryptographic schemes [1,2] provide cryptographically-secure privacy protections. However, not enough safeguards are provided in the anonymity revocation capability to prevent abuse. This problem can be attributed to a non user-centric approach. In a user-centric system, users - who are the owner of their information - should be *in control* of how their information is used [3]. The existing schemes [1,2] allow a user's identity (which is escrowed to an anonymity revocation manager (ARM)) to be revealed *without* the user's knowledge. There is technically nothing, except trust, to prevent the ARM from doing so. Nevertheless, to prevent abuse of an anonymity revocation capability, it is *essential* that users should know if their anonymity has been revoked to allow them to take immediate actions if such revocations have been misused. Although anonymity revocation can be linked to a set of conditions (for example, when a user has been diagnosed with a notifiable disease) which have to be fulfilled prior to the revocation [1,2], too much trust is placed on the ARM's honesty and ability to assess if such conditions are fulfilled.

Therefore, we argue that a protocol that enhances existing anonymity revocation schemes to provide the user-centric property, while decreasing the reliance on 'trust' in ARM. The contributions of this paper are: (1) identify the security requirements and the threat model for a user-centric anonymity revocation

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 185–194, 2008.

protocol (UC-ARP), and (2) the specification of UC-ARP based on the private credential system [1] and the custodian-hiding verifiable encryption scheme [4] which reduces the 'trust' requirement to only 1 out of t designated referees, and (3) its security and performance analysis. We henceforth call this protocol 1_{R_h}-UC-ARP. While key escrow forms a part of 1_{R_h}-UC-ARP, a user-centric approach is used (existing key-escrow proposals - such as [5]- are not user-centric).

This paper is organized as follows, section 2 describes the security requirements and the threat model for a user-centric anonymity revocation protocol. Section 3 provides some background information on the *existing* cryptographic schemes that are used in 1_{R_h}-UC-ARP. Section 4 provides the details of 1_{R_h}-UC-ARP. Section 5 briefly discusses how 1_{R_h}-UC-ARP has achieved the security requirements detailed in section 2, as well as the performance of 1_{R_h}-UC-ARP.

2 Security Requirements and Threats

The players in 1_{R_h}-UC-ARP are: Users (U) - entities whose anonymity is to be protected by this protocol, Providers (P)- entities that provide services which U consume, Referees (R) - entities that assess if a user's anonymity should be revoked, and the Anonymity Revocation Manager (ARM) - a designated referee who has been chosen by users as the entity who can finally, once a threshold is reached, revoke a user's anonymity.

Security Requirements. To our knowledge, no security properties specifically for anonymity revocation protocols have been proposed before. In [6], properties for a user-centric system are proposed, however, they are not specific enough for anonymity revocation purposes. Therefore, in this section, we extend [6] to document the specific properties for an anonymity revocation protocol.

Firstly, an anonymity revocation protocol has to be **user-centric**. To achieve this property, the protocol has to provide *user-knowledge* and *user-participation* properties. By *user-knowledge*, a user should know that his anonymity is to be revoked before it happens. This property is crucial in a user-centric system because it enables users to be in control of their anonymity and to take corrective actions if such revocation has been abused. *User-participation* can be *non-essential* where anonymity revocation is not dependent on the user participation, or *essential* where the revocation is dependent on the user participation. The *non-essential* user participation property is suitable for e-commerce where a user's anonymity can be revoked legitimately without his explicit participation as a malicious user may refuse to participate. The *essential* user-participation property is suitable in an environment involving highly sensitive personal information, such as health records where a user should give his explicit consent through participation in the protocol before his data can be accessed.

We also require an **enforceable conditions fulfillment** property, which can be direct or indirect: *Direct Enforcement of Conditions* means that anonymity revocation is dependent on actions which are **directly** related to conditions fulfillment. For example, in e-cash applications, action by a user who uses a coin twice results in the availability of enough data to revoke that user's anonymity.

Indirect Enforcement of Conditions means that fulfillment of a condition is determined by non-technical activities, however, once a referee is satisfied, actions can be performed (such as decrypting a ciphertext) that result in anonymity revocation. In other words, anonymity revocation is dependent on actions which are **indirectly** related to conditions fulfillment.

An anonymity revocation protocol should be **abuse resistant**: *ARM*s must not be able to revoke a user's anonymity without a *user-knowledge* and without leaving detectable traces. This property is important because if anonymity revocation capability is vulnerable to abuse, all of the other underlying privacy protections are rendered ineffective. Forcing *ARM*s to make a user 'aware' of such a revocation, and making them leave detectable traces will deter them from abusing such a capability.

Additionally, we also require an **authenticated user** property: while a user is anonymous prior to the revocation, the result of an anonymity revocation should correctly identify the intended user. A malicious user must not be able to masquerade as another user.

Finally, anonymity revocation requires an **events linkability** property once a user's anonymity is revoked. This property can either be *unlinkable events* where ARM and P must not be able to link the revoked identity with a set of anonymous past or future events conducted by the same user, or *linkable events* where ARM and P are able to link the revoked identity with a set of past events conducted by the same user. While *linkable events* may be more practical, *unlinkable events* provides a stronger privacy protection.

Threat Model. We assume the following threats for 1_{R_h}-UC-ARP protocol:

- Dishonest P who attempts to revoke users' anonymity whenever he can.
- Honest users U with a subset of dishonest users $U_{dh} \subset U$ who will attempt to masquerade as other users, and/or not cooperate in the revocation protocol.
- Dishonest ARM who will try to revoke users' anonymity without their knowledge as long as it does not leave 'traces' of its misbehavior. The ARM may provide a false identity of a user to P. If P intends to have the identity of a user u_a revoked, an ARM may provide the identity of $u_b \neq u_a$.
- Mostly honest referees ($R_h \subset R$) with a small subset of dishonest referees ($R_{dh} \subset R$, $R_h \cup R_{dh} = R$, $R_h \cap R_{dh} = \emptyset$) who can collude to revoke a user's anonymity without the user's knowledge. Referees want to protect their reputations, and decisions on conditions fulfillments are publicly known.

Collusion is possible between P, ARM, and R. Collusion between U and P, ARM, or R is unlikely due to their conflicting interests (U will want to remain anonymous, while P, ARM, R will want to revoke the users' anonymity).

3 Background Cryptographic Schemes

1_{R_h}-UC-ARP builds on the private credential framework proposed in [1] and the universal custodian-hiding verifiable encryption scheme (UCHVE) [4]. In this section, these schemes are explained at a high level to allow non-cryptography-specialist readers to understand the properties that they provide. Further details (algorithms, key requirements, proofs) can be obtained from the original papers.3

Notation: $m_a, m_b ... m_j$ are plain text data items.
$Cipher_{scheme-m_i} = Enc_{scheme}(m_i; K^i_{pub_{scheme}})$ is an encryption of a data item m_i using the Enc_{scheme} encryption scheme under i's public encryption key. The plain text can only be recovered by i who has the corresponding private key as input to decryption algorithm: $m_i = Dec_{scheme}(Cipher_{scheme-m_i}; K^i_{priv_{scheme}})$. $Cipher_{scheme-m_{i,j,k}}$ is a short form to refer to three separate encryptions, each containing encryption of data item m_i, m_j, and m_k respectively. A signature S_{m_i} over a message m_i can only be produced using i's private signing key: $S_{m_i} = Sign(m_i; K^i_{sign})$. Anybody can verify the signature using the public verification key of i: $VerifySign(S_{m_i}; m_i; K^i_{verify}) = 1$ (valid) or 0 (invalid).

A commitment c_{m_i} of a data item m_i is generated using a *Commit* algorithm, along with a random value r: $c_{m_i} = Commit(m_i, r)$. A commitment is *hiding*: it does not show any computational information on m_i, and *binding*: it is computationally impossible to find another $m_i^{'}$ and $r^{'}$ as inputs to the same *Commit* algorithm that gives a value $c'_{m_i} = c_{m_i}$. A hidden signature over a secret message m_i can be generated by knowing its corresponding commitment only: $S_{m_i} = HiddenSign(c_{m_i}; K^i_{sign})$.

$PK\{(m_a)$: $F(m_a, m_b ... m_i) = 1\}$ refers to a zero knowledge proof interactive protocol (PK). PK is executed between a Prover and a Verifier. The data on the left of the colon m_a is the data item that a Prover needs to prove the knowledge of such that the statements on the right side of the colon, $F(m_a, m_b ... m_i) = 1$, is correct. A verifier will not learn the data on the left hand side of the colon, while other parameters are known. The actual protocol involves one or more message exchange(s). At the end of the protocol, a verifier will be convinced (or not) that the prover has the knowledge of m_a without the verifier learning it.

Private Credential System: The private credential system (PCS) proposed in [1,2] is built upon several cryptographic schemes: the SRSA-CL (Strong RSA) signature scheme [7], Camenisch and Shoup verifiable encryption scheme [8] and BM-CL (Bilinear Mapping) signature scheme [9]. PCS provides many useful privacy-enhancing services, however, for the purpose of this paper, only the conditional anonymity revocation capability of this PCS is elaborated.

In PCS, *unlike* the 'usual' certificate (such as X509 certificate), a certificate $Cert_1$ issued to a user u_a is a signature of $CertificateIssuer_1$ over a collection of data items using either the SRSA-CL or BM-CL signature scheme: $Cert_1 = Sign(id_a, m_b, ... m_i; K^{CertificateIssuer_1}_{sign})$. A user u_a should keep $Cert_1$ *private*.

In this paper, we assume that the data item id_a in $Cert_1$ is the explicit ID of a user u_a, while $m_b, ... m_i$ contain other personal information (such as address, date of birth, etc). The anonymity revocation is accomplished as follows: id_a is *blinded* using a commitment scheme: $c_{id_a} = Commit(id_a, r)$. Then, the value id_a, hidden in c_{id_a}, is encrypted using the verifiable encryption scheme (VE) [8] under the *ARM* public encryption key, along with a set of pre-determined *Conditions*:

$Cipher_{VE-id_a} = Enc_{VE}(id_a; Conditions; K^{ARM}_{public-VE})$. Then, a PK is executed to prove that c_{id_a} is the commitment for id_a contained in $Cert_1$ issued by $CertificateIssuer_1$ (this is achieved by using the *proof of knowledge of a signature on committed messages* technique based on either the SRSA-CL or BM-CL signature scheme, depending on which signature scheme $Cert_1$ is generated - see [7,9] for details). This PK also proves that $Cipher_{VE-id_a}$ is an encryption of id_a hidden in c_{id_a}, under the ARM public key:

$$\begin{aligned} PK\{(Cert_1, id_a) \;:\; & c_{id_a} = Commit(id_a, r) \wedge \\ & VerifySign(id_a, m_b, .., m_i; K^{CertificateIssuer_1}_{verify}) = 1 \wedge \\ & Cipher_{VE-id_a} = Enc_{VE}(id_a; Conditions; K^{ARM}_{public-VE})\} \end{aligned} \quad (1)$$

Protocol (1) allows a user to provide a verifiable encryption of id_a without the verifier learning its value and still be convinced that $Cipher_{VE-id_a}$ contains id_a.[1] However, $Cipher_{VE-id_a}$ can be trivially decrypted by ARM without the user's knowledge, and there is no enforcement of *Conditions* fulfillment. Our UC-ARP protocol seeks to reduce the trust placed in the ARM.

Universal Custodian Hiding Verifiable Encryption (UCHVE): The UCHVE scheme [4] is used in 1_{R_h}-UC-ARP. Consider a discrete log relation: $y = g^x$ (y and g are known to verifier, but the discrete log value of x is private to a prover). For a group R of n referee members, the UCHVE encryption scheme allows a user to verifiably encrypt x to some designated t members from a subset group $T \subset R$ with any k out of these t members required to work jointly to recover x ($1 \le k \le t \le n$). T can be formed spontaneously and members of T can be different from session to session. The identities of the members of T are *hidden*. A *verifier* can only verify if the ciphertext received from a prover *correctly* encrypts x, in relation to the known value of y and g, and that any k members of T have to work together to recover x without learning the identity of the members of T, or the value of x. This encryption is denoted as $Enc_{UCHVE(k,t,n)}$. If $k = t$, that is, $Enc_{UCHVE(t,t,n)}$, then only when *all* t members of T work together can the encrypted message be recovered.

For members of T, t well-formed ciphertext pieces will be generated, each encrypted using the corresponding member of T's public keys. For members of R *not* in T, $n - t$ random values are chosen from specific domains such that they are indistinguishable from the well-formed ones. Regardless, there will be a total of n ciphertext pieces (well-formed + random). Intuitively, UCHVE scheme takes up substantial resources to perform, and therefore its use should be kept to a minimum.

To recover x (assuming that $k = t$), all members of R have to firstly verify that he/she is member of T by applying validation checking to the given ciphertext pieces (details in [4]). For members of R in T, such checking will be successful, and thus they can decrypt the given ciphertext and produce a *share*. For members of R *not* in T, such checking will be unsuccessful, thus output *reject* and stop. Once these t *shares* are collected, they are used as input to a particular *function*

[1] This protocol applies to any number of personal data items, not restricted to only id_a.

which will eventually output x. Any $t-1$ or less shares are not sufficient to recover x (see [4] for further details).

4 A New 1-Honest Referee UC-ARP ($\mathbf{1_{R_h}}$-UC-ARP)

The 1_{R_h}-UC-ARP detailed in this section satisfies the *user-knowledge, non-essential user-participation, abuse-resistant, indirect enforcement of conditions fulfillment, authenticated user, and unlinkable events* properties. A user is anonymous as long as id_a remains unrevealed (therefore, anonymity revocation only requires the revelation of one data item: id_a). Nevertheless, 1_{R_h}-UC-ARP can also be extended to protect a set of d data items ($d \geq 1$). 1_{R_h}-UC-ARP is divided into the *Setup, Identity Escrow, Key Escrow,* and *Revocation* stages.

Assumptions: 1_{R_h}-UC-ARP assumes that there is one honest referee out of t designated referees, and no collusion between users and referees. There is an existing public key infrastructure (PKI) that can be used. The details of the PKI infrastructure are *out of the scope* of this paper. The key-size, the encryption/signing algorithms and other parameters used are of sufficient length and strength to prevent reasonable adversaries from breaking the cryptographic schemes. A secure communication channel can be used as required.

Setup: U and R are grouped into one multicast group M. A user should obtain the necessary certificates from $CertificateIssuer_1$ as per the private credential system explained in section 3. For the purpose of this paper, consider that a user u_a has $Cert_1$ which P accepts as a source of the user's personal information. $Cert_1$ is verified using the verification key $K_{verify}^{CertifcateIssuer_1}$. $Cert_1$ contains $id_a, m_b, ...m_j$, with id_a as the explicit ID of u_a.

For $i = 1...n$, all referees form a group R of n members, and each has a set of publicly known encryption key $K^i_{pub_{UCHVE}}$ and the corresponding private key $K^i_{priv_{UCHVE}}$. It is assumed that u_a and P have agreed on a set of $Conditions$ before starting the protocol. The $Conditions$ should include a one-time unique value so that each set of $Conditions$ is unique.

Identity Escrow: This stage is similar to the one proposed in [1,2], with the exception that the user encrypts id_a using a one-time key, instead of the public key of ARM - see Figure 1.

1. The user u_a:
   (a) Generates a random number r, commit id_a: $c_{id_a} = Commit(id_a, r)$
   (b) Generates a one-time key pair for VE scheme [8]: $(K^u_{pub_{VE}}, K^u_{priv_{VE}})$
   (c) Encrypts id_a: $Cipher_{VE-id_a} = Enc_{VE}(id_a; Conditions; K^u_{public_{VE}})$
   (d) Sends $K^u_{pub_{VE}}$ and $Cipher_{VE-id_a}$ to P
2. u_a and P engage in PK to verify that c_{id_a} is a commitment of id_a from $Cert_1$, and $Cipher_{VE-id_a}$ is an encryption of id_a under $K^u_{pub_{VE}}$ (c_{id_a} will be made available to P as part of this PK)
3. u_a appoints a referee as the ARM, and sends $Cipher_{VE-id_a}$, $K^u_{pub_{VE}}$, and $Conditions$ to the ARM.

Identity Escrow:

U_a	P	ARM

$\xrightarrow{K^u_{pub_{VE}}, Cipher_{VE-id_a}}$

Execute PK with P: $PK\{(Cert_1, id_a) : c_{id_a} = Commit(id_a, r) \wedge$
$VerifySign(id_a, m_b, ..., m_i; K^{CertificateIssuer_1}_{verify} \wedge$
$Cipher_{VE-id_a} = Enc_{VE}(id_a; Conditions; K^u_{public_{VE}})$

$\xrightarrow{Cipher_{VE-id_a}, K^u_{pub_{VE}}, Conditions}$

Key Escrow:

U_a	ARM	P

$\xrightarrow{Cipher^{1...n}_{UCHVE-x_{1,2,3}}}$

Execute PK with ARM: $PK\{(x_1, x_2, x_3) : y_1 = g^{x_1}, y_2 = g^{x_2}, y_3 = g^{x_3} \wedge$
$Cipher^{1...n}_{UCHVE-x_{1,2,3}} = Enc_{UCHVE}(x^{1-n}_{1,2,3}; Conditions; K^{referee_i}_{public-UCHVE})$

$\xleftarrow{S_{Cipher_{VE-id_a},Conds}, S_{Receipt}, Receipt}$ Sends $\xrightarrow{S_{Cipher_{VE-id_a},Conds}, S_{Receipt}, Receipt}$

Fig. 1. 1_{R_h}-UC-ARP - Identity and Key Escrow

4. Optionally, P can perform hidden signature on c_{id_a}:
   $S_{id_a} = HiddenSign(c_{id_a}; K^P_{sign})$, and link S_{id_a} with $Cipher_{VE-id_a}$

The ARM now has a verified ciphertext of id_a which it cannot decrypt for it does not have the private key $K^u_{priv_{VE}}$. This key is escrowed in the next stage.

Key Escrow: As per the key specifications of the VE scheme [8], $K^u_{priv_{VE}}$ is composed of three components: x_1, x_2, x_3, each one of them is related to the public keys in a discrete log relationship: $y_1 = g^{x_1}$, $y_2 = g^{x_2}$, $y_3 = g^{x_3}$ (g, $x_{1,2,3}$ and *several other* public key components are chosen according to the key generation algorithm detailed in [8]). Each $x_1, x_2,$, and x_3 is verifiably encrypted to the designated members of $T \subset R$ using the $Enc_{UCHVE(t,t,n)}$ scheme.

1. u_a spontaneously forms T with t members $(t > 1)$ out of n referees in R.
2. u_a encrypts x_1, x_2, x_3 for members of T using UCHVE (t, t, n) scheme - under the same *Conditions* as the ones used in the identity escrow stage. For $i \in T$, well-formed ciphertext pieces are generated:
   $Cipher^i_{UCHVE(t,t,n)-x_{1,2,3}} =$
   $Enc_{UCHVE(t,t,n)}(x^i_{1,2,3}; Conditions; K^{referee_i}_{public_{UCHVE}})$.
   For $i \in [1, n] \backslash T$, random values chosen accordingly so that they are indistinguishable from the well-formed ones, giving a total of n ciphertext pieces.
3. u_a sends $Cipher^{1...n}_{UCHVE-x_{1,2,3}}$ to ARM
4. u_a and ARM engage in a PK protocol to prove that the collection of $Cipher^{1...n}_{UCHVE-x_{1,2,3}}$ encrypt x_1, x_2, and x_3 under the same *Conditions*, and that all referees in T have to jointly recover each of these components without learning the identities of the members of T. This is straight forward as x_1, x_2, x_3 are the discrete log values of the publicly known y_1, y_2, y_3.

5. The ARM stores $Cipher^{1...n}_{UCHVE-x_{1,2,3}}$, and links this to $Cipher_{VE-id_a}$ and $Conditions$. Then ARM signs:
   - $S_{Cipher_{VE-id_a},Conditions} = Sign(Cipher_{VE-id_a}, Conditions; K^{ARM}_{sign})$
   - $S_{Receipt} = Sign(Receipt;K^{ARM}_{sign})$ ($Receipt$ = statement that the encrypted id_a, conditions, and encrypted private key are received)
6. ARM sends $S_{Cipher_{VE-id_a},Conditions}$, $S_{Receipt}$ and $Receipt$ to P and u_a.
7. u_a and P verify $S_{Cipher_{VE-id_a},Conditions}$ and $S_{Receipt}$ to ensure that ARM has the correct encrypted id_a and $Conditions$.

$Enc_{UCHVE(t,t,n)}$ scheme is chosen because it provides a *verifiable* encryption of discrete log values and it hides the identity of members of T (this property is exploited in the Revocation stage). In section 5, a discussion will be provided on how *the number of referees in T and how T is formed by u_a* are crucial in determining the security of this protocol.

It may be possible to use $Enc_{UCHVE(t,t,n)}$, instead of the VE scheme [8], in the identity escrow stage, and remove the key escrow stage. However, while this is doable when we only need to encrypt one data item id_a, it becomes *unscalable* once we encrypt many data items. If $d > 1$, the UCHVE scheme will result in dn ciphertexts (d refers to the number of data items to be encrypted, n for the number of referees in R). The value of d can be greater than 1 due to the following: (1) The UCHVE scheme *restricts the length* of a data item that is to be encrypted to be smaller than a certain length l, thus, if the length of id_a (or any data items) is greater than l, then id_a has to be split into several data items, hence $d > 1$, and (2) we may need to escrow several of u_a's personal information, in addition to id_a.

With key escrow, there will always be *at most* $3n$ resulting ciphertexts encrypted using UCHVE scheme (3 for the three secret keys $x_{1,2,3}$), irrespective of the value of d. To increase the efficiency further, it may possible to escrow only x_1, (x_2 and x_3 can be given to the ARM at later point without the escrow process as possession of x_2 and x_3 is *insufficient* to decrypt $Cipher_{VE-id_a}$). Therefore, we can have only n escrowed-key ciphertext pieces, regardless of d. The above key escrow protocol escrows all three private key components.

Revocation: The anonymity revocation procedure is as follows:

1. P asks ARM to revoke u_a's anonymity by sending $S_{Cipher_{VE-id_a},Conditions}$ to ARM
2. The ARM multicasts $mesg = S_{Cipher_{VE-id_a},Conditions} + Conditions$ to the multicast group M ($M = U + R$) to indicate that u_a's anonymity is about to be revoked. Each user and referee should verify if $mesg$ is sent through the multicast address for M. If not, ARM must have misbehaved. Stops.
3. Each users checks if they have the same $S_{Cipher_{VE-id_a},Conditions}$ value stored. The user u_a must have the same value stored from the key escrow stage. Therefore, u_a knows that his/her anonymity is being revoked.
   - The user u_a sends the identities of the referees in T ($id - referee_{1...t}$) to ARM. The ARM sends $Cipher^{j}_{UCHVE-x_{1,2,3}}$ to each $referee_j \in T$ ($j = 1...t$).

- However, u_a can also refuse to send $id-referee_{1...t}$ to ARM. If so, the ARM sends $Cipher^j_{UCHVE-x_{1,2,3}}$ to *all* $referee_j \in R$ $(j = 1...n)$

4. Each $referee_j$ verifies $Conditions$ received from step 2 are satisfied (methods vary and are out of the scope of the protocol). If so, applies check to the given $Cipher^j_{UCHVE-x_{1,2,3}}$ to verify if he/she is member of T. For $referee_j \in T$, such checking will be successful, thus the referee can proceed to decrypt $Cipher^j_{UCHVE-x_{1,2,3}}$ using its private key under the same $Conditions$, resulting in valid *shares* $P^j_{1,2,3}$. Sends $P^j_{x_{1,2,3}}$ to ARM. For $referee_j \notin T$, such checking will fail, thus stops.
5. If ARM receives all shares $P^{1...t}_{x_{1,2,3}}$, it can recover x_1, x_2, x_3 $(K^u_{priv_{VE}})$, decrypt $Cipher_{VE-id_a}$ to get id_a, and send id_a to P. P can optionally generate S'_{id_a}, and verify $S'_{id_a} = S_{id_a}$ (generated at identity escrow stage).
6. If ARM does not get all of the required $P_{1...t}$, anonymity revocation fails. As we assume there is one honest referee $r_h \in T$, revocation will fail unless $Conditions$ are fulfilled and $mesg$ is sent through the multicast address.

An ARM will skip step 2 if it wants to revoke u_a's anonymity without the user knowledge for u_a will not obtain $mesg$ if it is not sent through the multicast address. As T is formed spontaneously by u_a during the key escrow stage, the members of T **vary** from session to session. Therefore, the ARM will be compelled to send the $mesg$ in step 2 because it needs to know $id-referee_{1...t}$ from u_a so that it can optimize performance by only sending t ciphertext pieces to members of T. Granted, u_a may refuse to reveal $id-referee_{1...t}$. However, if $Conditions$ are fulfilled, id_a will *eventually* be revealed and it is better for the user u_a to cooperate earlier to avoid being 'black-listed' for being non-cooperative.

5 Discussion

The proposed 1_{R_h}-UC-ARP achieves the security properties as detailed in section 2 provided that there is at least one honest referee $r_h \in T$ and that the underlying cryptographic schemes are secure and correct. Specifically, the 1_{R_h}-UC-ARP satisfies the following properties: *user-knowledge, non-essential user-participation, abuse-resistant, indirect enforcement of conditions fulfillment, authenticated user, and unlinkable events.* Please refer to the extended version of our paper at [10] for the detailed discussion on how these properties are achieved.

Performance. The performance of the 1_{R_h}-UC-ARP is analyzed based on the number of modular exponentiation (*modEx*) required. Table 1 provides the number of *additional modEx* operation required (as compared to the existing approach in [1, 2]) for both the maximum (escrowing x_1, x_2, x_3 without user cooperation) and optimum (escrowing only x_1 with user cooperation during revocation) cases. The details of how we obtain such figures are provided in [10].

Clearly, the bottleneck is at the ARM due to the significant number of *modEx* operations required in PK-UCHVE [4]. As mentioned in section 4, the performance of 1_{R_h}-UC-ARP depends on the value of t (we assume n to be constant).

Table 1. Additional *online* modular exponentiation required for 1_{R_h}-UC-ARP

	User	**ARM**	n **referees**	**Provider**	Total
Maximum	$6t + 6n + 2$	$3t + 39n + 2$	$6n + 6t$	2	$15t + 51n + 6$
Optimum	$2t + 2n + 2$	$t + 13n + 2$	$4t$	2	$7t + 15n + 6$

In step 1 of the key escrow stage, a large t chosen by user means more *modEx* required, lowering performance. 1_{R_h}-UC-ARP reaches its worst performance when $t = n$, however, it is also when the protocol is most secure as we now only require 1 honest referee out of n referees. Such trade-offs between performance and security are inevitable.

Furture work includes increasing the efficiency of 1_{R_h}-UC-ARP, strengthening the protocol to be secure even when users and referees collude, and research into how to achieve the *Direct Enforcement of Conditions* property.

References

1. Bangerter, E., Camenisch, J., Lysyanskaya, A.: A cryptographic framework for the controlled release of certified data. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2004. LNCS, vol. 3957, pp. 20–42. Springer, Heidelberg (2006)
2. Camenisch, J., Sommer, D., Zimmermann, R.: A general certification framework with applications to privacy-enhancing certificate infrastructures. In: Fischer-Hübner, S., Rannenberg, K., Yngström, L., Lindskog, S. (eds.) SEC. IFIP, vol. 201, pp. 25–37. Springer, Heidelberg (2006)
3. Brands, S.: Identity: Setting the larger context, achieving the right outcomes. In: 7th Annual Privacy and Security Workshop & 15th CACR Information Security Workshop (November 2006)
4. Liu, J.K., Tsang, P.P., Wong, D.S., Zhu, R.W.: Universal custodian-hiding verifiable encryption for discrete logarithms. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 389–409. Springer, Heidelberg (2006)
5. Bellare, M., Goldwasser, S.: Verifiable partial key escrow. In: 4th ACM CCS, pp. 78–91. ACM, New York (1997)
6. Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D.: User centricity: a taxonomy and open issues. In: Juels, A., Winslett, M., Goto, A. (eds.) DIM, pp. 1–10. ACM, New York (2006)
7. 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)
8. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
9. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
10. Suriadi, S., Foo, E., Smith, J.: A user-centric protocol for conditional anonymity revocation. Technical Report 13123, Queensland University of Technology (March 2008), `http://eprints.qut.edu.au/archive/00013123/`

Preservation of Privacy in Thwarting the Ballot Stuffing Scheme

Wesley Brandi, Martin S. Olivier, and Alf Zugenmaier

Information and Computer Security Architectures (ICSA) Research Group
Department of Computer Science, University of Pretoria, Pretoria

Abstract. Users of an online trading system rely on Reputation Systems to better judge whom should be trusted and to what degree. This is achieved through users building up reputations in the system. In these types of environments, it has been shown that users with good reputations do more business than users with bad reputations. The ballot stuffing scheme exploits this and has fraudulent users placing a number of false bids in an attempt to better the reputation of a single user.

Though previous research has dealt with thwarting the one man ballot stuffing scheme, the issue of privacy was neglected. The solution proposed relied on looking up the coordinates of a user who is a cellular phone holder. Upon placing a bid, the user's geographical coordinates are compared to the coordinates of other users involved in the transaction. If the users were within a predefined distance to one another, the transaction was marked as suspicious. This mechanism relies on storing the coordinates of a user over time and, from a privacy perspective, is unacceptable.

The intention of this paper is to propose several solutions that attempt to safeguard the privacy of all users involved when calculating the distance between two cellular phone holders, i.e., thwarting the one man ballot stuffing scheme. We discuss solutions that cater for service providers who may be willing or unwilling to participate in safeguarding the privacy of their users. These techniques include Secure Multi-party Computation, polynomial interpolation and the addition of untrusted third parties.

1 Introduction

In the absence of the normal social interactions one associates with doing business face to face, online trading systems must rely on other mechanisms to assist users in mitigating the risk that may come from trading with unknown entities. Reputation systems [14] have been shown to be an effective mechanism to achieve exactly this purpose. With a reputation system in place, users of an online trading system have more information at hand to assist in deciding whom to trust and to what degree.

Users with a strong reputation stand a better chance of conducting more business than users with a weak reputation [15]. It is upon this premise that incentive for the ballot stuffing scheme is born. Users wishing to strengthen their

S.M. Furnell, S.K. Katsikas, and A. Lioy (Eds.): TrustBus 2008, LNCS 5185, pp. 195–204, 2008.

reputation may collude with multiple buyers in a number of fake transactions. Upon completion of each transaction the colluding buyers will rate the seller in a manner that increases his reputation.

Previous research [1] dealt with the one man ballot stuffing scheme and a means for thwarting this scheme using cell phones. Cell phones were introduced to the system and were a prerequisite to making a bid. Upon placing a bid, the coordinates of the associated cell phone were looked up and compared to that of previous buyers. If the phone was within a predetermined distance from any of the other bidders, or the seller, then the bid was marked as suspicious (the assumption being that it was highly unlikely for bidders in a global trading system to be geographically close to one another).

A downfall of this framework is the issue of privacy. Gorlach *e*t al [8] point out that revealing the location of individuals can be a serious threat to their privacy. Scenarios are discussed where AIDS patients could be identified by the offices of the doctors that they visit or religious groups by the churches they frequent. Storing the geographical coordinates of all users placing bids in a trading system, albeit an approach to preventing ballot stuffing, has similar privacy implications.

In this paper we address these privacy concerns. Specifically, we discuss six techniques that allow the distance between two parties to be measured (or at least compared) in a privacy-preserving manner. Each of the techniques proposed has a different set of assumptions. These assumptions draw from a variety of areas which include collaborating service providers, accuracy of results, trusted third parties and communication overheads.

This paper is structured as follows: section 2 discusses related work in the fields of Secure Multi-party Computation and privacy preserving distance measurement. In section 3 we discuss the motivation behind this research. Section 4 then provides three solutions to preserving privacy when thwarting the one man ballot stuffing scheme. This section assumes that the service providers involved are willing to collaborate with one another. Section 5 discusses another three solutions but assumes that the service providers are not willing to collaborate with another. In section 6 we conclude this paper.

2 Background

Yao [19,20] introduces the notion of Secure Multi-Party Computation with a discussion of the Millionaires Problem. Two millionaires decide that they want to know who is the richer of the two. The problem is that neither of them wishes to tell the other how much money they have. The protocol proposed by Yao makes some assumptions: (1) the two millionaires agree on what the upper and lower bound of their wealth is and (2) both millionaires employ the usage of public key infrastructure.

Goldreich *e*t al [7] generalise the problem of Secure Multi-party Computation (SMC) and employ circuit evaluation gates to prove that there exists a method to calculate any $f(x_1, .., x_i)$ for i parties in a private and secure fashion. Additional general solutions to the SMC problem are proposed in [3,13]. With a simple enough function, these general solutions can be practical. In most cases though, as the number of participants involved in the computation grows and as the complexity of the function to be computed increases, the general solutions become impractical [6]. As a result, optimised SMC solutions to a number of problems in varying domains are of interest and have received attention [10,16,9,18].

Du [5] and Vaidya [17] discuss an approach to SMC that has a significant impact on the performance of the calculations performed. Du argues that an ideal approach to security is not always necessary. It may be the case that people won't mind sacrificing some security (just how much security is adjustable) if the gain in performance is substantial, i.e., if the trade off makes SMC a practical alternative. Vadya's approach is similar and has a security trade off in the form of a participating and untrusted third party.

Of particular interest in this paper are SMC solutions that have been developed to support computational geometry. This is a relatively new field and has received initial attention from Du and Atallah [4] in their approach to the following geometric problems: point inclusion, intersection, closest pair and convex hulls.

Recently, Yonglong *e*t al [12] and Li *et al* [11] studied the problem of calculating the distance between two points within the domain of SMC. The solution proposed by Li incorporates the 1-out-of-m Oblivious Transfer protocol [2]: assuming party B has m inputs $X_1, X_2, ..., X_m$, the 1-out-of-m protocol allows party A to choose a single X_i where $1 \leq i \leq m$. Party B does not know which input was chosen (he does not know i).

3 Motivation

A responsible trading system must respect the fact that storage of a patron's geographical coordinates is private and confidential. Even in the case where the system has the consent of the individual to store the details, the trading system must employ whatever means possible to safeguard his/her privacy.

The framework proposed to thwart ballot stuffing is designed to operate as follows:

1. A bidder makes a bid in a transaction. The framework already knows the cellular number of the bidder (through registration) and has his permission to look up his location (the location of the device).
2. The location of the cell phone is looked up (via the appropriate service provider) and stored in the trading system database.
3. This location is then compared to the locations of recent bidders related to the current transaction/seller. If the distances between the bidders is within a certain range, the transaction is tagged accordingly

The distance between bidders was initially thought to be imperative to the framework in the sense that the framework needed to know the actual value. What previous research omitted though, is that with the help of the cellular service providers, the trading system may not need to know the geographical coordinates or the actual distances between the bidders at all. Cooperation from the service providers is imperative in this case.

The rest of this paper is structured on the following two scenarios:

> Collaborating service providers - in this scenario cellular service providers are willing to collaborate with one another in addition to the trading system. It is their responsibility to safeguard the privacy of each of their patrons. As a result the geographical coordinates of each patron must be protected from other service providers; this includes third parties the likes of a trading system.
>
> Uncooperative service providers - it may be the case that cellular service providers are not interested in the overhead incurred when collaborating with other parties. They will provide the geographical coordinates of their users (provided that each user has given consent) and the onus will then fall upon the trading system to safeguard the privacy of the user.

4 Collaborating Service Providers

The problem being dealt with in this paper can be summarised as follows: there are three parties involved of which two are cellular service providers and one is a trading system. The trading system simply wants to determine the distance between two cellular phones (where each phone belongs to one of the service providers). How does the trading system determine the distance between the phones by not infringing on the privacy of the phone holder?

In this section we discuss solutions that safeguard the privacy of the cell phone holders through collaborating service providers. By collaborating with one another, the trading system (and any other party for that matter) is able to perform its necessary calculations without exposing the coordinates of any user at any time.

4.1 Solution I - Calculating Distance

The ideal solution to this problem has the coordinates of a user protected not only from the trading system but from collaborating service providers as well. In solving the problem, neither of the parties involved must be able to infer any of the coordinates held by either service provider.

Using the SMC distance measurement technique proposed by Li and Dai [11], at least one of the service providers will know the distance between the two patrons. In following this protocol, neither of the two service providers will know anything other than the geographical coordinates of *their* patron and the computed distance to the other service provider's coordinates.

Two problems are immediately apparent: (1) what can be inferred by knowing the distance in addition to knowing only one of the coordinates and (2) how does each service provider refer to the geographical coordinates of a different service provider's coordinates over a period of time?

The latter problem applies to the trading system and is more of an implementation issue. Fortunately, it is easily solved in the form of a ticketing system. Details handed out by the service provider may include the coordinates in addition to a unique identifier (the ticket) that may be used to look up the associated coordinate at a later stage, for example, the coordinates for the cell phone u_1 may be released by the service provider in the form (x, y, t_1). At a later stage the (x, y) coordinates may be retrieved using only the ticket t_1.

The former problem has the potential to be far more serious. In calculating the distance from a patron of Service Provider 1 (SP_1) to a patron of Service Provider 2 (SP_2), the service provider now has a basis from which to start inferring exactly where the patron of SP_2 might be.

Initially, there may be too many possibilities from which to draw a meaningful conclusion, i.e., the cell phone holder could be located on any point on the circumference of a circle with a radius of n kilometers. However, if SP_1 applies the knowledge of its patron's coordinates in addition to domain specific knowledge it may have regarding SP_2 it may be in a position to make far better decisions as to where SP_2 might be.

4.2 Solution II - Comparing Distance

In the previous section we discuss the implications of privately calculating the distance from one service provider's patron to another. Unfortunately, knowing the distance to another service provider's coordinates in addition to one's own coordinates can be used to launch a privacy attack.

Fortunately, the privacy-preserving distance measurement protocol developed by Yonglong *e*t al [12] allows two parties to compute the distance from two points without either of the parties actually knowing what the distance is. Essentially, the two parties are left with a result that can be used to compare the distance to something else. The authors discuss a point inclusion problem which is closely related to the problem discussed in this paper.

Alice and Bob want to determine if Alice's point $p(x_0, y_0)$ falls within Bob's circle $C : (x - a)^2 + (y - b)^2 = r^2$. As is always the case in these examples, Alice and Bob do not want to share their data with one another. In protocol 4 of their paper (point-inclusion), Yonglong *e*t al describe a technique whereby Bob and Alice collaborate with one another to determine if Alice's point falls within Bob's circle.

The process consists of two steps. In the first step they use the technique developed in protocol 3 (two dimensional Euclidean distance measure protocol) which leaves Alice with $s = n + v$ where n is the actual distance from the center of Bob's circle to Alice's point and v is a random variable known only to Bob. The two parties then use the millionaire protocol to compare variants of s in order to establish whether or not Alice's point is within Bob's circle.

This example is important because no coordinates were shared between the two parties. In addition to this, the distance between the two points (n) was compared to r using a protocol where neither party knew n and only one party knew r. By extending the problem discussed in this paper from a distance problem to a point inclusion problem we can preserve the privacy of all parties concerned and prevent attacks of the nature described in the previous section.

Changing the problem to that of a point-inclusion problem is trivial: the trading system simply wants to determine if the distance between SP_1 (Bob's C) and SP_2 (Alice's p) is within a predefined distance r (specified by the trading system).

Note that although the point-inclusion protocol is proposed as an efficient way to calculate distance in a privacy preserving fashion there is still a significant overhead in its computation.

4.3 Solution III - An Untrusted Third Party

A simpler solution with less overhead involves the addition of an untrusted third party. The trading system is untrusted in so far as neither of the service providers trust it enough to provide it with the coordinates of its patrons. It is also the reason why the computations are being performed at all. By association, the trading system is therefore a perfect candidate for an untrusted third party.

As an untrusted third party, the service providers will provide a skewed version of their coordinates in a manner that has the trading system computing the real distance between the coordinates but not being able to infer the coordinates themselves. The solution proposed relies on minimal collaboration between the service providers and an arbitrary computation from the trading system's point of view.

Before hand, the service providers collaborate amongst themselves to decide on a random x_0 and y_0 that will be used to skew their coordinates. Both service providers then skew their coordinates $(x + x_0, y + y_0)$ before sending them off to the trading system. The obvious impact of this operation is that the points have been equally displaced in a manner that retains the distance between them. Since the trading system will never know x_0 or y_0 the service providers are safe from an inference attack.

This approach can be generalized to any distance preserving affine transformation of the coordinate system, i.e., a combination of rotation by angle θ, translation by vector (x_0,y_0) and reflection given by σ which can take the values of $+1$ and -1.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \sigma x - x_0 \\ y - y_0 \end{pmatrix} \tag{1}$$

The transformation parameters could even be made to depend on the identities of the pair of users. To enable this, the trading system would have to include the identity of the bidder of which it wants the coordinates and a hash of the identity of the other bidder in the request. The service providers would need to agree beforehand on a function to derive the parameters of the coordinate transformation from the bidder identities.

This approach assumes that the trading system (or any of the users) is not colluding with either of the service providers. If the trading system is colluding with one of the service providers then an attack is trivial since the x_0 and y_0 values will be known.

5 Uncooperative Service Providers

In this section we discuss solutions whereby the service providers are assumed to be uncooperative, i.e., they are not willing to collaborate with any third party (this includes other service providers and the trading system) in order to preserve the privacy of their users. Essentially, the role of preserving the privacy of the users is shifted to the third party (in our case this is the trading system) upon receiving the coordinates of users (with their consent) from the service provider.

Of course, shifting the preservation of privacy to the trading system is somewhat of a problem. The previous section dealt with protecting the privacy of a user by not allowing anyone other than the originating service provider to know his coordinates. Surely by discussing solutions where these coordinates are sent to the trading system we are contradicting ourselves? If the trading system is not a trusted third party then this is certainly the case. No untrusted third party can be privy to private information through which the coordinates of a person are revealed over some period of time.

With this problem in mind, we define a specific notion of a privacy violation in this section: the privacy of a user is said to have been violated if any party (other than those who have been given explicit consent) is able to accurately retrieve/infer the user's coordinates.

Though the trading system may be viewed upon as a trusted third party, the preservation of a user's privacy is by no means trivial. Users' coordinates could be stored in a database maintained by the trading system but in the event of it being compromised, the trading system is then responsible for a massive privacy violation.

In this section we will discuss several solutions that look at mechanisms the trading system could employ to safeguard a user's privacy in addition to protecting its own interests in the event of being compromised.

5.1 Solution IV - Probable Privacy

The concept behind this solution is that of data loss. Upon receiving coordinates (x_1, y_1), the trading system uses them until the point where it wishes to persist them. Only a downgraded version of the coordinates will be stored. In our case, the y coordinate is thrown away (lost) and only the x coordinate is saved. More generally a projection Π of the coordinates on any one-dimensional linear subspace can be used.

In the event of the trading system being compromised, the only data that an attacker will have to work with is that of x coordinates. On the other hand, as much as the attacker has to deal with x coordinates, the trading system must somehow deal with them as well.

Remember that the trading system does not need to know the actual distance between two cell phones. Rather, it needs to know if the distance between two cell phones is less than a certain threshold. In comparing a new (x, y) coordinate to a set of old x coordinates the trading system can employ simple mathematics to discard x coordinates that are obviously not candidates for what it is looking for.

x coordinates which have not been discarded now need to be assessed in so far as what the probability is that the $l$ost y coordinate was within a region that makes the distance between the coordinates fit within a predetermined threshold. Though this seems impractical when one considers a coordinate system the size of planet earth and a distance threshold of a few hundred meters, this may be viable if the technique discussed to attack solution I is adopted, i.e., that of applying additional knowledge to the system. This could include attaching meta data to x coordinates, for example, the service provider of x_3 only operates within y coordinates a to b.

5.2 Solution V - Polynomial Interpolation

In this section we extend the data loss principle discussed in solution IV. This is achieved in the form of data recovery using a polynomial function. We know that given n points in the form of (x_i, y_i) coordinates, there exists a polynomial p to the $n - 1$ degree such that $p(x_i) = y_i$ (assuming every x_i is unique).

This is applied to the trading system by generating a new p each time a new coordinate is received. Since all the x coordinates are stored already, p will be used to generate all the y coordinates. The new coordinates will then be included when generating the new p. Once this has been completed, all y coordinates are then discarded. If the trading system then stores the polynomial p in addition to the x coordinates then the y coordinates of a user will be recoverable when necessary.

Obviously, having p and x_i accessible by the trading system at any time is only changing the nature of the problem slightly. Ideally, p and x_i should not be held by the same entity. If p were given to a trusted third party (trusted by the trading system) then y_i would only be accessible when both parties are collaborating.

An alternative to a third party involves the service provider. The service provider is already being used to provide coordinates of a user at some point in time. The trading system could exploit this service and encrypt p with a key that only the service provider would have, i.e., one of the coordinates of the user (remember that the trading system $l$oses all y coordinates). When the trading system needs to access p then it would ask for the key from the service provider, for example, by sending a request for the n^{th} coordinate of a particular user.

Note that in the beginning of this section we pointed out that all x_i coordinates must be unique. If this solution is to be considered at all then there can be no duplicate x_i coordinates. This is easily addressed in the form of a trade off, i.e., shifting duplicate x_i coordinates by one or two units.

5.3 Solution VI - Data Partitioning

The solution proposed in this section is a combination of the previous two solutions and includes the following:

> Data Massaging - before the y coordinate is lost we construct an interval of adjustable length denoting where the y coordinate may have been, for example, if y was 17 we may store the following: [15,21].
> Partitioning - The y interval is then stored with a trusted third party.

Solution IV is effective in so far as discarding irrelevant x coordinates. However, when processing x coordinates that may be relevant, the overheads of applying additional knowledge to the probability calculations in order to determine y may be substantial. With this in mind, Solution V makes y available through a polynomial that may be encrypted or stored with a third party, y is therefore made available when it is required and as a result both coordinates are available at some point in time.

From a privacy perspective this may be a problem. Although the coordinates are not revealed all of the time, they are available some of the time. By massaging the y coordinate into an interval rather than an explicit value in addition to partitioning the data, the trading system may be able to make more accurate judgements with less overheads in a more privacy centric fashion.

6 Conclusion

This paper has presented six solutions to calculate (or compare) the distance between two coordinates over time in a privacy-preserving fashion. These solutions offer to protect the privacy of a user's coordinates from a third party the likes of a trading system in addition to other service providers.

Choosing the best solution depends on the type of environment in which it will be implemented. The solutions proposed in this paper have been offered in an attempt to provide a range of choices when safeguarding the privacy of individuals in a trading system. Note that although the context of this paper is within the realm of a trading system and two service providers, the solutions themselves need not be limited to this environment alone.

References

1. Brandi, W., Olivier, M.S.: On bidder zones, cell phones and ballot stuffing. In: Proceedings of Information Security South Africa (ISSA) (July 2006)
2. Brassard, G., Crépeau, C., Robert, J.: All-or-nothing disclosure of secrets. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234–238. Springer, Heidelberg (1987)
3. Chaum, D., Crépeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: STOC 1988: Proceedings of the twentieth annual ACM symposium on Theory of computing, pp. 11–19. ACM, New York (1988)

4. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications: a review and open problems. In: NSPW 2001: Proceedings of the 2001 workshop on New security paradigms, pp. 13–22. ACM Press, New York (2001)
5. Du, W., Zhan, Z.: A practical approach to solve secure multi-party computation problems. In: NSPW 2002: Proceedings of the 2002 workshop on New security paradigms, pp. 127–135. ACM Press, New York (2002)
6. Goldreich, O.: Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, New York (2004)
7. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the nineteenth annual ACM conference on Theory of computing, pp. 218–229. ACM, New York (1987)
8. Gorlach, A., Heinemann, A., Terpstra, W.: Survey on location privacy in pervasive computing. In: Robinson, P., Vogt, H., Wagealla, W. (eds.) Privacy, Security and Trust within the Context of Pervasive Computing. The Kluwer International Series in Engineering and Computer Science (2004)
9. Ioannidis, I., Grama, A., Atallah, M.: A secure protocol for computing dot-products in clustered and distributed environments. In: ICPP 2002: Proceedings of the 2002 International Conference on Parallel Processing (ICPP 2002), Washington, DC, USA, p. 379. IEEE Computer Society Press, Los Alamitos (2002)
10. Kiltz, E., Leander, G., Malone-Lee, J.: Secure computation of the mean and related statistics. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 283–302. Springer, Heidelberg (2005)
11. Li, S., Dai, Y.: Secure two-party computational geometry. J. Comput. Sci. Technol. 20(2), 258–263 (2005)
12. Luo, Y., Huang, L., Chen, G., Shen, H.: Privacy-preserving distance measurement and its applications. Chinese Journal of Electronics 15(2), 237–241 (2006)
13. Naor, M., Nissim, K.: Communication preserving protocols for secure function evaluation. In: STOC 2001: Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 590–599. ACM Press, New York (2001)
14. Friedman, E., Resnick, P., Zeckhauser, R., Kuwabara, K.: Reputation systems. Communication of the ACM 43(12), 45–48 (2000)
15. Resnick, P., Zeckhauser, R., Swanson, J., Lockwood, K.: The value of reputation on ebay: A controlled experiment (2003)
16. Shundong, L., Tiange, S., Yiqi, D.: Secure multi-party computation of set-inclusion and graph-inclusion. Computer Research and Development 42(10), 1647–1653 (2005)
17. Vaidya, J., Clifton, C.: Leveraging the "multi" in secure multi-party computation. In: WPES 2003: Proceedings of the 2003 ACM workshop on Privacy in the electronic society, pp. 53–59. ACM Press, New York (2003)
18. Vaidya, J., Clifton, C.: Secure set intersection cardinality with application to association rule mining. Journal of Computer Security 13(4), 593–622 (2005)
19. Yao, A.: Protocols for secure computations (extended abstract). In: Proceedings of FOCS 1982, pp. 160–164 (1982)
20. Yao, A.: How to generate and exchange secrets. In: Proceedings of FOCS 1986, pp. 162–167 (1986)

Author Index