Joaquin Garcia-Alfaro
Guillermo Navarro-Arribas
Nora Cuppens-Boulahia
Yves Roudier (Eds.)

LNCS 5939

Data Privacy Management and Autonomous Spontaneous Security

4th International Workshop, DPM 2009 and Second International Workshop, SETOP 2009 St. Malo, France, September 2009, Revised Selected Papers

Springer

Joaquin Garcia-Alfaro
Guillermo Navarro-Arribas
Nora Cuppens-Boulahia
Yves Roudier (Eds.)

Data Privacy Management and Autonomous Spontaneous Security

4th International Workshop, DPM 2009
and Second International Workshop, SETOP 2009
St. Malo, France, September 24-25, 2009
Revised Selected Papers

Springer

Volume Editors

Joaquin Garcia-Alfaro
Nora Cuppens-Boulahia
TELECOM Bretagne, Campus de Rennes
2, rue de la Châtaigneraie, 35512 Cesson Sévigné, Cedex, France
E-mail: {joaquin.garcia, nora.cuppens}@telecom-bretagne.eu

Guillermo Navarro-Arribas
IIIA-CSIC, Campus UAB, 08193 Bellaterra, Spain
E-mail: guille@iiia.csic.es

Yves Roudier
Institut Eurécom
2229 Route des Crêtes - BP 193, 06904 Sophia Antipolis Cedex, France
E-mail: yves.roudier@eurecom.fr

Library of Congress Control Number: 2009941441

CR Subject Classification (1998): E.3, K.4.1, K.4.4, K.6.5, C.3, D.4.6, C.2

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-642-11206-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-11206-5 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12826629 06/3180 5 4 3 2 1 0

Foreword from the Program Chairs of DPM 2009

Organizations are increasingly concerned about the privacy of information that they manage (several people have filed lawsuits against organizations violating the privacy of customers' data). Thus, the management of privacy-sensitive information is very critical and important for every organization. This poses several challenging problems, such as how to translate the high-level business goals into system-level privacy policies, administration of privacy-sensitive data, privacy data integration and engineering, privacy access control mechanisms, information-oriented security, and query execution on privacy-sensitive data for partial answers.

The 4th International Workshop on Data Privacy Management (DPM) was the continuation of the International Workshop on Privacy Data Management, which held three previous issues (2005 in Tokyo, Japan; 2006 in Atlanta, USA; and 2007 in Istanbul, Turkey). After one year of inactivity the workshop started again in 2009 in Saint Malo, France, co-located with the ESORICS conference. And plans are to continue the workshop on a yearly base.

The Program Committee accepted for presentation 8 papers out of 23 submissions from 13 different countries in four continents. Each submitted paper received at least three reviews. These proceedings contain the revised versions of these papers, covering topics such as privacy in service-oriented architectures, privacy-preserving mechanisms, cross-matching and indistinguishability techniques, privacy policies, and disclosure of information. The workshop also had two keynote speakers. Josep Domingo-Ferrer, from Universitat Rovira i Virgili, and Chairman of the UNESCO Chair in Data Privacy; and Tomas Sander, from the Systems Security Lab of Hewlett-Packard Labs in Princeton.

Many people deserve our gratitude. We would like to thank the General Chairs of DPM 2009 (Josep Domingo-Ferrer and Vicenç Torra), and the General Chairs of ESORICS 2009 (Frederic Cuppens and Nora Cuppens-Boulahia). The Organizing Committee from ESORICS 2009 helped with the local organization. We thank all the Program Committee members for their help, availability, and commitment. In evaluating the papers submitted we received the help of the Program Committee and four external reviewers (Junfeng Fan, Joan Melia-Segui, Alfredo Rial, and Joerg Schneider). In the same vein we thank the sponsors of the workshop for helping with financial and logistical issues: The UNESCO Chair in Data Privacy, the Spanish-founded projects ARES-CONSOLIDER CSD2007-00004, and e-Aegis TSI2007-65406-C03-01, the Artificial Intelligence Research Institute (IIIA) from the Spanish National Research Council (CSIC), the Internet Interdisciplinary Institute (IN3) of the Open University of Catalonia (UOC), and the IEEE Spanish Section. We also thank the help of Mukesh Mohania, Jordi Castella-Roca, and Julien A. Thomas. Finally, we extend our thanks to Katell Himeur for her invaluable support during the organization of the event.

September 2009

Joaquin Garcia-Alfaro
Guillermo Navarro-Arribas

Foreword from the Program Chairs of SETOP 2009

SETOP is a companion event of the ESORICS symposium which presents research results on all aspects related to spontaneous and autonomous security. This year, the second issue of SETOP was held in St. Malo, a beautiful walled port city in Brittany in north-western France during September 24-25, 2009.

With the need for evolution, if not revolution, of current network architectures and the Internet, autonomous and spontaneous management will be a key feature of future networks and information systems. In this context, security is an essential property. It must be considered at the early stage of conception of these systems and designed to also be autonomous and spontaneous. Future networks and systems must be able to automatically configure themselves with respect to their security policies. The security policy specification must be dynamic and adapt itself to the changing environment. Those networks and systems should interoperate securely when their respective security policies are heterogeneous and possibly conflicting. They must be able to autonomously evaluate the impact of an intrusion in order to spontaneously select the appropriate and relevant response when a given intrusion is detected.

Autonomous and spontaneous security is a major requirement of future networks and systems. Of course, it is crucial to address this issue in different wireless and mobile technologies available today such as RFID, Wifi, Wimax, 3G, etc. Other technologies such as ad hoc and sensor networks, which introduce new types of services, also share similar requirements for an autonomous and spontaneous management of security.

The high quality of SETOP 2009 papers facilitated a stimulating exchange of ideas among the members of the international research community interested in this crucial topic of spontaneous and autonomous security. SETOP 2009 was honored to have three distinguished keynote speakers – Peng Ning from North Carolina State University, Josep Domingo-Ferrer from Universitat Rovira i Virgili Catalonia and Roberto di Pietro from Università Roma 3. Thank you, Peng, Josep and Roberto for having accepted our invitation.

We are also very grateful to our sponsors: DCSSI, INRIA, Rennes Métropole, Région Bretagne, Fondation Métivier, Saint-Malo, Alcatel-Lucent Bell Labs France, EADS, Orange, TELECOM Institute, CG35. Their generosity has helped keep the costs of SETOP 2009 moderate.

September 2009

Nora Cuppens-Boulahia
Yves Roudier

4th International Workshop on Data Privacy Management – DPM 2009

Program Committee Chairs

Joaquin Garcia-Alfaro	UOC/TELECOM Bretagne
Guillermo Navarro-Arribas	IIIA-CSIC

Workshop General Chairs

Josep Domingo-Ferrer	Universitat Rovira i Virgili
Vicenç Torra	IIIA-CSIC

Program Committee

Alessandro Acquisti	Carnegie Mellon University
Michel Barbeau	Carleton University
Marina Blanton	University of Notre Dame
Joan Borrell	Autonomous University of Barcelona
Iliano Cervesato	Carnegie Mellon University
Valentina Ciriani	University of Milan
Frédéric Cuppens	TELECOM Bretagne
Nora Cuppens-Boulahia	TELECOM Bretagne
Ernesto Damiani	University of Milan
Claudia Diaz	K.U.Leuven-Heverlee
Josep Domingo-Ferrer	Rovira i Virgili University
David Evans	University of Cambridge
Joaquin Garcia-Alfaro	UOC/TELECOM Bretagne
Stefanos Gritzalis	University of the Aegean
Jordi Herrera	Autonomous University of Barcelona
Apu Kapadia	MIT Lincoln Laboratory
Evangelos Kranakis	Carleton University
Loukas Lazos	University of Arizona
Kun Liu	IBM Almaden Research Center
Fabio Massacci	Universita di Trento
Gero Muhl	Berlin University of Technology
Guillermo Navarro-Arribas	IIIA-CSIC
Radha Poovendran	University of Washington
Utz Roedig	Lancaster University
Thierry Sans	Carnegie Mellon University
Vicenç Torra	IIIA-CSIC
Nicola Zannone	Eindhoven University of Technology

Organizing Committee

Joaquin Garcia-Alfaro	UOC/TELECOM Bretagne
Guillermo Navarro-Arribas	IIIA-CSIC
Josep Domingo-Ferrer	Universitat Rovira i Virgili
Vicenç Torra	IIIA-CSIC

Additional Referees

Junfeng Fan	K.U.Leuven, ESAT/COSIC
Joan Melia-Segui	Open University of Catalonia
Alfredo Rial	K.U.Leuven, ESAT/COSIC
Joerg Schneider	Berlin University of Technology

Second International Workshop on Autonomous and Spontaneous Security – SETOP 2009

Program Committee Chairs

Nora Cuppens-Boulahia	TELECOM Bretagne, Rennes
Yves Roudier	EURECOM, Sophia-Antipolis

General Chair

Evangelos Kranakis	Carleton University, Ottawa

Organization Chair

Frédéric Cuppens	TELECOM Bretagne, Rennes

Program Committee

Michel Barbeau	Carleton University, Ottawa
Christophe Bidan	Supélec, Rennes
Ana Cavalli	TELECOM SudParis, Evry
Hakima Chaouchi	TELECOM SudParis, Evry
Claude Chaudet	TELECOM ParisTech, Paris
Yves Correc	DGA/CELAR, Bruz
Frédéric Cuppens	TELECOM Bretagne, Rennes
Hervé Debar	France Télécom R&D, Caen
Jose M. Fernandez	École Polytechnique de Montréal
Noria Foukia	University of Otago, New Zealand
Alban gabillon	Université polynesie Française
Joaquin Garcia-Alfaro	Carleton University, Ottawa
Evangelos Kranakis	Carleton University, Ottawa
Loukas Lazos	University of Arizona, Tucson
Jean Leneutre	TELECOM ParisTech, Paris
Javiez lopez	University of Malaga
Maryline Maknavicius	TELECOM SudParis, Evry
Catherine Meadows	Naval Research Laboratory
Refik Molva	EURECOM, Sophia-Antipolis
Radha Poovendran	University of Washington, Seattle
Juan Carlos Ruiz	UPV, Valencia
Thierry Sans	Carnegie Mellon, Doha

Organizing Committee

Nora Cuppens-Boulahia	TELECOM Bretagne, Rennes
Frédéric Cuppens	TELECOM Bretagne, Rennes
Gilbert Martineau	TELECOM Bretagne, Rennes (Sponsor Chair)
Julien Thomas	TELECOM Bretagne, Rennes

Table of Contents

Keynote Talks

Data Privacy Management

Autonomous and Spontaneous Security

The UNESCO Chair in Data Privacy Research in Vehicular Networks

Josep Domingo-Ferrer

Universitat Rovira i Virgili,
UNESCO Chair in Data Privacy,
Dept. of Computer Engineering and Mathematics,
Av. Països Catalans 26, E-43007 Tarragona, Catalonia
josep.domingo@urv.cat

Abstract. An overview of the activities of the UNESCO Chair in Data Privacy is first given. One of these activities is research. We focus on the research conducted to conciliate security and privacy in vehicular ad hoc networks (VANETs) and, specifically, in VANET announcements.

Keywords: Vehicular *ad hoc* networks, Privacy, Trust, Car-to-car messages.

1 Introduction

The UNESCO Chair in Data Privacy (http://unescoprivacychair.urv.cat) is an agreement between UNESCO and Universitat Rovira i Virgili, who acts as a host institution for the Chair. The agreement was signed on March 6, 2007, and it is renewed every two years by mutual consent. A UNESCO Chair must do research, cooperation, training and dissemination in a field considered relevant by UNESCO for the welfare of humankind; in the case of the Chair in Data Privacy, the focus is on privacy, already mentioned as a fundamental right in Article 12 of the Universal Declaration of Human Rights (1948):

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against such interference or attacks.

Beyond the host institution, there are participating institutions in the UNESCO Chair in Data Privacy including, among others, the United Nations Economic Commission for Europe (UN/ECE), CSIC (Spain's Higher Council for Scientific Research), Sabanci University (Istanbul, Turkey), Destatis-Statistisches Bundesamt (Germany) and CBS-Statistics Netherlands.

The most visible actions by the Chair include:

Dissemination: Organization of the biennial *Privacy in Statistical Databases-PSD* conference, with LNCS proceedings (Barcelona, 2004, LNCS 3050;

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 1–8, 2010.

Rome, 2006, LNCS 4302; Istanbul, 2008, LNCS 5262; Corfu, 2010) and publication of the *Transactions on Data Privacy* journal (TDP, http://www.tdp.cat). TDP is jointly published with IIIA-CSIC and it is currently indexed by DBLP, ACM Digital Library, MathScinet and DOAJ.

Co-operation: The Chair regularly sponsors a number of privacy research conferences by offering travel grants for authors and attendees from transition countries.

Research Researchers from the Chair co-ordinate several research projects on creating new information technologies that conciliate privacy, security and technology. The most revelant of those is the CONSOLIDER INGENIO 2010 project "ARES" (http://crises-deim.urv.cat/ares), a five-year endeavor (2007-2012) co-ordinated by this author and involving a multinational team of about 80 researchers from six different universities.

In the rest of this talk, we focus on a specific research scenario where we are particularly active at the Chair's host institution (URV): vehicular *ad hoc* networks (VANETs). It will be argued that VANETs are especially challenging in what regards the combination of privacy, security and functionality. Section 2 introduces VANETs. Section 3 reviews the countermeasures proposed in the literature to obtain secure and privacy-preserving VANETs. Section 4 discusses how to combine *a priori* and *a posteriori* countermeasures in order to overcome the shortcomings of proposals in the literature. Section 5 is a conclusion.

2 Vehicular *Ad Hoc* Networks

According to recent technology forecasts [1], vehicles will be equipped with radio interfaces in the near future and vehicle-to-vehicle (V2V) communications will be available in vehicles by 2011. The IEEE 802.11p task group is working on the Dedicated Short Range Communications (DSRC) standard which aims at enhancing the 802.11 protocol to support wireless data communications for vehicles and the road-side infrastructure [19]. Car manufacturers and telecommunication industry gear up to equip each car with devices known as On-Board Units (OBUs) that allow vehicles to communicate with each other, as well as to supply Road-Side Units (RSUs) to improve safety, traffic efficiency, driver assistance, and transportation regulation. The RSUs are expected to be located at the critical points of the road, such as traffic lights at road intersections. The OBUs and RSUs form a self-organized network called a VANET, emerging as the first commercial instantiation of the mobile *ad hoc* networking (MANET) technology.

VANETs allow nodes including vehicles or road-side infrastructure units to communicate with each other over single or multiple hops. In other words, nodes will act both as end points and routers. Vehicular networking protocols allow vehicles to broadcast messages to other vehicles in the vicinity. It is suggested that each vehicle periodically send messages over a single hop every 300ms within a distance of 10s travel time (which means a distance range between 10m and 300m)[17]. This mechanism can be used to improve safety and optimize traffic.

However, malicious vehicles can also make use of this mechanism by sending fraudulent messages for their own profit or just to jeopardize the traffic system. Hence, the system must be designed to ensure that the transmission comes from a trusted source and has not been tampered with since transmission.

Another critical concern in VANETs is the privacy or anonymity of the driver (or the vehicle, for that matter). As noted in [6], a lot can be inferred about the driver if the whereabouts and the driving pattern of a car can be tracked. It is indeed possible for attackers to trace vehicles by using cameras or physical tracking, but such physical attacks can only trace specific targets and are much more expensive than monitoring the communication in VANETs. Hence, most studies focus on thwarting the latter attacks.

3 Countermeasures for Securing VANETs

VANETs can improve traffic safety only if the messages sent by vehicles are trustworthy. Dealing with fraudulent messages is a thorny issue for safety engineers due to the self-organized operation of VANETs. The situation is further deteriorated by the privacy requirements of vehicles since, in a privacy-preserving setting, the message generators, *i.e.* the vehicles, are anonymous and cannot be identified when acting maliciously. A number of schemes have been proposed to reduce fraudulent messages; such proposals fall into two classes, namely *a posteriori* and *a priori*.

3.1 *A Posteriori* Countermeasures

A posteriori countermeasures consist in taking punitive action against vehicles which have been proven to have originated fraudulent messages. To be compatible with privacy preservation, these countermeasures require the presence of a trusted third party able to open the identities of dishonest vehicles. Then the identified vehicles can be removed from the system.

Cryptographic authentication technologies have been extensively exploited to offer *a posteriori* countermeasures. Most proposals use regular digital signatures and require a public-key infrastructure. See [5] for a survey.

A critical issue posed by vehicular message authentication is driver's privacy. Since the public key used to verify the authenticated messages can be linked to specific users, attackers can trace vehicles by observing vehicular communications. Hence, mechanisms must be adopted to guarantee vehicle/driver privacy when vehicles authenticate messages. Along this research line, there are two main approaches: pseudonymous mechanisms and group signatures.

In a pseudonymous mechanism, the certificate authorities produce multiple pseudonyms for each vehicle so that attackers cannot trace the vehicles producing signatures in different periods under different pseudonyms, except if the certificate authorities open the identities of the vehicles. Pseudonymous mechanisms have been extensively investigated from various aspects. Short-lived certificates are also suggested in [11], mainly from the perspective of how often a node should change a pseudonym and with whom it should communicate. The authors of [18] propose to use a silent period in order to hamper linkability between pseudonyms, or alternatively to create groups of vehicles and restrict vehicles in one group from hearing messages of other groups.

This conditional anonymity of pseudonymous authentication will help determining the liability of drivers in the case of accidents. The downside of this approach is the necessity for generation, delivery, storage, and verification of numerous certificates for all the keys. To mitigate this heavy overhead, [2] presents an approach to enable vehicle on-board units to generate their own pseudonyms without interacting with the CAs. The mechanism is realized with the help of group signatures. In [10] a novel group signature-based security framework is proposed which relies on tamper-resistant devices (requiring password access) for preventing adversarial attacks on vehicular networks. However, they provide no concrete instantiation or experiment analysis.

In [12], the authors propose a secure and privacy-preserving protocol for VANETs by integrating the techniques of group signature and identity-based signature. In their proposal, they take into account security and privacy preservation between OBUs, as well as between OBUs and RSUs. In the former aspect, a group signature is employed to secure the communication between OBUs, where messages are anonymously signed by the senders while the identities of the senders can be traced by the trusted authorities if the messages are later found to be doubtable. In the latter aspect, an identity-based signature scheme is used at RSUs to sign each message generated by RSUs to ensure its authenticity. With their approach, the heavy load of certificate management can be greatly reduced.

3.2 *A Priori* Countermeasures

VANETs can improve traffic safety and efficiency only if vehicular messages are correct and precise. Despite the security provided by the combination of TPDs with authenticated messages, an attacker could still manage to transmit valid messages containing false data. It is easy for an attacker to launch such an attack. For instance, putting the vehicle temperature sensor in cold water will let the OBUs generate false messages, even if the hardware sensors are tamper-proof. Also, one may note that in some cases the sender of the data may not necessarily be malicious, but his vehicle's sensors may be out of order. To rule out such cases of false data, one needs not only to verify that the sender of the data is legitimate, but also that the data are correct. Therefore some mechanisms for detection of malicious data need to be explored. We refer to such approaches as *a priori* countermeasures which attempt to prevent the generation of erroneous messages in advance.

A detailed survey on *a priori* countermeasures can be found in [5]. Here we will mention only the most efficient class, namely threshold-based mechanisms [8,14,15,16,4]. In these proposals, a message is trusted only if it was endorsed by a number of vehicles in the vicinity. This approach is based on the assumption that most users are honest and will not endorse any message containing false data. Another implicit assumption is the usual common sense that, the more people endorse a message, the more trustworthy it is. Among these schemes, the proposals in [4] may be the most efficient while enabling anonymity of message originators by exploiting secret sharing techniques. But their scheme does not provide anonymity revocability, which may not suit some applications in which anonymity must be revoked "for the prevention, investigation, detection and prosecution of serious criminal offences" [7].

3.3 Discussion on Existing Countermeasures

Unfortunately, neither *a posteriori* nor *a priori* countermeasures suffice on their own to secure VANETs. By taking strict punitive action, *a posteriori* countermeasures can protect against rational attackers producing bogus messages to obtain benefits or pranks. However, they are ineffective against irrational attackers such as terrorists. Even for rational attackers, damage has already occurred when punitive action is taken. It seems that *a priori* countermeasures function better in this case because they prevent damage beforehand by letting the vehicles trust only messages endorsed by a certain number of vehicles. However, although the underlying assumption that there is a majority of honest vehicles in VANETs generally holds, it cannot be guaranteed that a number of malicious vehicles greater than or equal to the threshold will never be present at specific locations. For example, this is likely to happen if some criminal organization undertakes to divert traffic from a certain area by broadcasting messages informing that a road is barred. Furthermore, for convenience of implementation, existing schemes use an even stronger assumption that the number of honest vehicles in all cases should be at least a preset threshold. But such a universally valid threshold does not exist in practice. Indeed, the threshold should somehow take the traffic density and the message scope into account: a low density of vehicles calls for a lower threshold, whereas a high density and a message relevant to the entire traffic of a city requires a sufficiently high threshold.

The situation is aggravated by the anonymity technologies used in some proposals. A system preserves anonymity when it does not require the identity of its users to be disclosed. Without anonymity, attackers can trace all the vehicles by monitoring the communication in VANETs, which in turn can enable the attackers to mount serious attacks against specific targets. Hence, anonymity is a critical concern in VANETs. However, anonymity can also weaken *a posteriori* and *a priori* countermeasures. Indeed, attackers can send fraudulent messages without fear of being caught, due to anonymity; as a result, no punitive action can be taken against them. Furthermore, some proposals provide strong anonymity, *i.e.* unlinkability. Unlinkability implies that a verifier cannot distinguish whether two signatures come from the same vehicle or two vehicles. This feature may enable malicious vehicles to mount the so-called Sybil attack: a vehicle generates a fraudulent message and then endorses the message herself by computing on it as many signatures as required by the threshold in use; since signatures are unlinkable, no one can find out that all of them come from the same vehicle. Hence, elegantly designed protocols are required to secure VANETs when incorporating anonymity. It must be noted that, among those threshold-based systems cited above which provide *a priori* protection and anonymity, [4] is the only one resistant to the Sybil attack: in that system, vehicles belong to groups, and vehicles in a group share keys (which provides vehicle anonymity because vehicles in a group are interchangeable as far as signing goes); however, for a message to be validated, endorsements from a number of different groups are needed, so a single vehicle cannot get a message sufficiently endorsed.

4 Towards a Combination of *a Priori* and *a Posteriori* Countermeasures

Our focus is to devise a context-aware threshold authentication framework with conditional privacy in VANETs, equipped with the following properties: i) it should be privacy-preserving; ii) it should support an adaptive threshold authentication mechanism (*a priori* security); iii) it should allow anonymity revocation in case of offence (*a posteriori* security).

4.1 Message-Linkable Group Signatures

Group signatures have been investigated for many years [3,9]. In a group signature scheme, each group member can anonymously sign messages on behalf of the group. However, a group manager can open the identity of the author of any group signature in case of dispute. Most existing group signatures provide unlinkability in the sense that no efficient algorithm can tell whether two group signatures are generated by the same group member, even if the two signatures are on the same message. Linkable group signatures [13] are a variant of group signatures. In a linkable group signature, it is easy to identify the group signatures produced by an identical signer, even if the signer is anonymous. This feature is desirable in e-voting systems where each voter can anonymously vote only once.

Group signatures are useful for securing VANETs but they are vulnerable to the Sybil attack because of unlinkability. Linkable group signatures can thwart the Sybil attack but are not compatible with vehicle privacy due to the linkability of signer identities, *i.e.* the various message endorsements signed by a certain vehicle can be linked. Hence, a more sophisticated notion of linkability is required in group signatures for VANETs. Motivated by this observation, we presented in [5,20] a new primitive referred to as message-linkable group signatures (MLGS).

An MLGS scheme has the same security properties as regular group signatures except that, given two signatures on *the same message*, one can easily decide whether the two signatures are generated by the same member or by two different members, but the originator(s) stay(s) anonymous.

4.2 A New Solution Based on Message-Linkable Group Signatures

Based on MLGS, we propose a general framework for threshold authentication with revocable anonymity in VANETs. In this framework, each vehicle registers to a vehicle administration office serving as a group registration manager. When t vehicles wish to endorse some message, they can independently generate an MLGS signature on that message. After validating t MLGS signatures on the message, the verifying vehicle is convinced by the authenticated message. However, if later the message is found incorrect, the police office as well as judges (serving as the tracing manager) can trace the t cheating signers. Here, we assume that an honest signer never needs to sign the same message twice. This assumption is workable by embedding a time-stamp in each message, as suggested in most authentication schemes for VANETs, if the OBU of a vehicle senses the same situation at different times.

From the security properties of MLGS schemes, it is clear that the above framework satisfies the required properties of privacy preservation, as well as *a priori* and *a posteriori* security. If $t-1$ vehicles produce t signatures on the same message, then there exists a group member who has been involved in generating at least two signatures. Such an impersonation can be easily identified since the MLGS scheme is message-linkable. Furthermore, the resulting scheme is highly efficient, as required in VANETs. See [20] for more details.

5 Conclusions

We have presented the activities of the UNESCO Chair in Data Privacy and we have sketched the state of the art and the Chair research as regards security and privacy in vehicular networks.

Acknowledgments and Disclaimer

This work was partly supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-0004 “ARES” and TSI2007-65406-C03-01 “E-AEGIS”, and by the Government of Catalonia under 2009 SGR 1135. The author is partly supported as an ICREA-Acadèmia researcher by the Government of Catalonia. He is with the UNESCO Chair in Data Privacy, but his views do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. Blau, J.: Car talk. IEEE Spectrum 45(10), 16 (2008)
2. Calandriello, G., Papadimitratos, P., Lioy, A., Hubaux, J.-P.: Efficient and robust pseudonymous authentication in VANET. In: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks-VANET 2007, pp. 19–28 (2007)
3. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
4. Daza, V., Domingo-Ferrer, J., Sebe, F., Viejo, A.: Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology 58(4), 1876–1886 (2009)

5. Domingo-Ferrer, J., Wu, Q.: Safety and privacy in vehicular communications. In: Bettini, C., Jajodia, S., Samarati, P., Wang, S. (eds.) Privacy in Location-Based Applications, ch. 3. LNCS, vol. 5599, pp. 173–189. Springer, Heidelberg (2009)
6. Dötzer, F.: Privacy issues in vehicular ad hoc networks. In: Danezis, G., Martin, D. (eds.) PET 2005. LNCS, vol. 3856, pp. 197–209. Springer, Heidelberg (2006)
7. European Parliament. Legislative resolution on the proposal for a directive of the European Parliament and of the Council on the retention of data processed in connection with the provision of public electronic communication services and amending Directive 2002/58/EC (COM(2005)0438 C6-0293/2005 2005/0182(COD)) (2005)
8. Golle, P., Greene, D., Staddon, J.: Detecting and correcting malicious data in VANETs. In: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, pp. 29–37 (2004)
9. Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)
10. Guo, J., Baugh, J.P., Wang, S.: A group signature based secure and privacy-preserving vehicular communication framework. In: Mobile Networking for Vehicular Environments, pp. 103–108 (2007)
11. Jakobsson, M., Wetzel, S.: Efficient attribute authentication with applications to ad hoc networks. In: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks - VANET 2004 (2004)
12. Lin, X., Sun, X., Ho, P.-H., Shen, X.: GSIS: A secure and privacy preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology 56(6), 3442–3456 (2007)
13. Nakanishi, T., Fujiwara, T., Watanabe, H.: A linkable group signature and its application to secret voting. Transactions of Information Processing Society of Japan 40(7), 3085–3096 (1999)
14. Ostermaier, B., Dötzer, F., Strassberger, M.: Enhancing the security of local danger warnings in VANETs - A simulative analysis of voting schemes. In: Proceedings of the Second International Conference on Availability, Reliability and Security, pp. 422–431 (2007)
15. Parno, B., Perrig, A.: Challenges in securing vehicular networks. In: Proceedings of the ACM Workshop on Hot Topics in Networks (2005)
16. Raya, M., Aziz, A., Hubaux, J.-P.: Efficient secure aggregation in VANETs. In: Proceedings of the 3rd International Workshop on Vehicular Ad hoc Networks - VANET 2006, pp. 67–75 (2006)
17. Raya, M., Hubaux, J.-P.: The security of vehicular ad hoc networks. In: 3rd ACM Workshop on Security of Ad hoc and Sensor Networks-SASN 2005, pp. 11–21 (2005)
18. Sampigethaya, K., Huang, L., Li, M., Poovendran, R., Matsuura, K., Sezaki, K.: CARAVAN: Providing Location Privacy for VANET. Proc. of ESCAR 2005 (November 2005)
19. U.S. Department of Transportation, National Highway Traffic Safety Administration, Vehicle Safety Communications Project, Final Report (April 2006), http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/060419-0843/PDFTOC.htm
20. Wu, Q., Domingo-Ferrer, J., González-Nicolás, Ú.: Balanced trustworthiness, safety and privacy in vehicle-to-vehicle Communications. IEEE Transactions on Vehicular Technology (to appear, 2009)

Privacy Management for Global Organizations

Siani Pearson[1], Tomas Sander[2], and Rajneesh Sharma[3]

[1] Systems Security Lab, HP Labs, Filton Rd, Bristol, UK
[2] Systems Security Lab, HP Labs, 5 Vaughn Dr, Princeton, USA
[3] GBS BCP and Security Team, Wind Tunnel Rd, Bangalore, India
{Siani.Pearson,Tomas.Sander,Rajneesh.Sharma}@hp.com

Abstract. In this paper we look at the complex area of a global outsourcing delivery model among different countries and/or organizations. In this case, privacy requirements stemming from requirements of various countries of data origin need to be honoured and taken into account during the data lifecycle. We review practical privacy management challenges arising in large, global organizations and discuss technology needed to address them. As a first example we describe the design of a privacy tool built and deployed to help an organization identify and manage privacy concerns in the context of Business Process Outsourcing (BPO). As a generalization of this technology we present an automated solution for scalable, accountable privacy management.

Keywords: Privacy management, business process, outsourcing, accountability.

1 Motivation

Privacy management for multinational companies is challenging due to the complex web of legal requirements, distributed business activities and movement of data and business operations to cost-effective locations. Privacy requirements need to be addressed by numerous dispersed teams, within the context of a variety of business processes. Moreover, within a business process privacy requirements vary at different stages of the process. Known technical point solutions such as encryption technologies and auditing tools often address only a small part of overall privacy requirements; decision support tools are needed to provide assistance in the management of privacy knowledge, policies, requirements and controls.

In this paper we describe two tools that provide the abovementioned assistance:

- *BPO Privacy Manager (PM)*: this focuses on the global outsourcing delivery model
- *Accountability Model Tool (AMT)*: this captures data about business processes to determine their privacy compliance

2 BPO Lifecycle and Requirements

To understand privacy management requirements within a BPO organization we analysed differing privacy concerns within the BPO data lifecycle. A summary of the

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 9–17, 2010.

results is shown in Table 1. A key insight from this analysis is that privacy for companies is about *managing privacy requirements end-to-end*, and this motivates the need for tools that provide assistance for implementing privacy policies, requirements and controls. These need to be useful primarily to humans and not to machines. A first solution to this issue is described in the following section.

Table 1. Different Privacy Concerns within a Deal Lifecycle

Deal Pursuit	Due Diligence	Contract	Transition	Ramp Up	Go Live
Privacy concerns • High level assessments, e.g. : •Deal spoilers/ Show stoppers •Transborder data flow issues •Assessment of privacy risk and indication of need for detailed legal clearance •Indicators of expensive solutions for HP •Overall deal privacy risk rating ***Input available at this stage:*** •High level data such as Countries, industry type, customer requirements etc	***Privacy concerns*** •Detailed analysis of applicable privacy requirements •Determination of which responsibilities are HP's and which of the customer •Determination of privacy related clauses, disclaimers etc. to be put in the contract •Identification of technical components to be put into the solution •Identification of privacy related cost drivers •Levels of privacy protection required for solution ***Input available:*** •Detailed info about data to be processed •Applications used •Customer preferences	The contract is created and negotiated from the guidelines of the due diligence stage and the contact is signed	***•Privacy concerns*** •Specifications of technical and procedural guidelines •Determination of what needs to be done • Training of employees	***Privacy concerns*** •Measurement of effectiveness of technical and procedural privacy measures •Continuous improvement	***Privacy concerns*** •Monitoring •Reporting •Ongoing compliance checking •Prevention of privacy violations •Incident Management

3 BPO Privacy Manager (PM)

In this section we present our solution that addresses these requirements and issues in the BPO context.

We decided to focus on BPO as a domain initially, even though similar techniques would be used more broadly, for example in application services outsourcing or IT outsourcing, as a continuation of previous collaboration. We provided decision support for the first two stages of the BPO deal lifecycle (cf. Table 1) because provision of advice during these phases minimizes legal risk and knowledge and learning is built upon as the deal progresses. We included a 'learning capture' module that enables peer to peer knowledge transfer to capture the learning of tool users based on their real life experiences of dealing with privacy requirements. This also ensured that the learnings are not lost even though tool users leave the organization. We encoded the advice from privacy experts into our Knowledge Base (KB). We investigated a variety of different approaches for knowledge representation, including production rule systems, expert systems and a simpler database (DB) matching technique. Since the latter was able to produce the desired output, and reduced the knowledge management and maintenance overhead, we chose this approach.

BPO PM provides feedback to BPO operatives about privacy in BPO deal scenarios (see Figure 1). Using the deal information that the BPO users provide through the user interface (UI), the tool can flag upfront any privacy concerns from its knowledge, list privacy requirements and suggest controls to meet these requirements, and also show relevant advice from other users. The system has a KB about requirements and legislation; since this type of information changes over time, the tool also provides a management interface that allows privacy experts to update the KB so that that the BPO users have up-to-date information.

The user at the top right hand side of Figure 1 would interact with the systems via a series of UIs. When the user clicks on these screens, the deal information is captured and sent to a Java engine, where reasoning is carried out. The results of the reasoning are returned to the user (Figure 2 is one example). Deal information and historical information is stored in a DB, This need not be on the same machine as the engine – in general, the back end can be fully distributed.

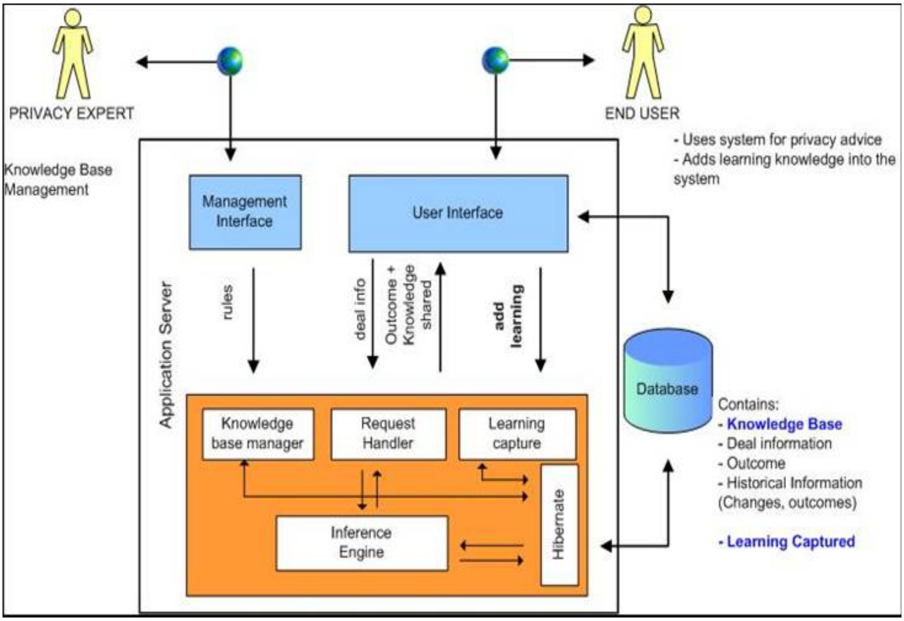


Fig. 1. Architecture of BPO Privacy Manager

PM takes into consideration the business context: we implemented interfaces to capture the knowledge and relate it to business conditions in which the knowledge should be shared and trigger it by comparing the business context with these conditions. In this way the knowledge is shared only when someone faces the same business condition to which the knowledge was related.

The data model can be separated into two distinct parts:

1. *Knowledge Base (KB):*

   – DP rules: these specify specific values of input parameters associated with flag values and reasons (strings) that are mapped to remediation (strings) such that if a match occurs with the current deal input parameters, they can trigger a flag to be (re)set to a certain value and/or reasons and remediation to be added to the output.

– DD rules: these map specific input values to requirements, requirements to controls and controls to associated comments (learning).

2. *Deal related information:* associated with each deal reference.

The rules for the DP phase cover information privacy rules at a high to medium level of detail and may vary according to the country of origin of the data. These include: transborder data flow regulations; handling of sensitive data; rules about call recordings; specific country alerts about the treatment of certain type of data or sectors. A meta-level description of the privacy rules for this phase is:

```
'if <trigger conditions> then <output>'
```

where the trigger conditions state matching values of predefined deal characteristics, and the output is composed of a flag (that can be red, yellow or green) that provides the overall privacy-related risk rating of the deal, the reasons for this flag and pointers towards remediations that can be taken to address this issue, if applicable. Remediations can be for example procedural measures, contact details for people that should be contacted, technological measures, etc.

In our implementation we use a database representation where Privacy Requirement Nodes (PRNs) are defined that are composed of database entries consisting of the text of the requirements (i.e. a brief description of a policy or legislative aspect), and the parameter values under which the requirement will be triggered. The parameters include data classification, data type, destination, type of destination country, industry type, source, way of collection, type of outsourcing, etc. This PRN is mapped in the DB to more detail about the source, including related web links, and to the flag and reasons.

Values can include groups. Two different types of exception may be allowed for within the value expressions:

- a group apart from one or more members of that group (e.g. when defining a PRN for EU countries except France)
- a group plus one or more countries that are not member of that group (e.g. when defining a PRN for EU countries that also applies to Canada)

This database representation only allows for trigger conditions formed from conjunctions of parameter value pairs or their negations. The representation does not allow more complex logical trigger conditions. If disjunctions within trigger conditions are needed, two separate rules (one for each case) have to be defined.

For example, a transborder data flow requirement could be:

"If the data transfer is to India and the deal involves PII the flag should be yellow". The PRN could have values of India as data exporter or data controller, and PII (personally identifiable information) as the type of data to be outsourced. The corresponding output for this PRN could be:

a. **Flag**: *Yellow*
b. **Flag description**: *Transborder data flow can be a concern*

c. **Reason**: *India is considered as a non-adequate country for the transfer of personal information from the EU/EEA*

d. **Remediations:**

- *The data exporter and the data importer/processor need to have a model-contract in place. <further details>.*
- *Data transfer notification must usually be made to the Data Privacy Agency in the country where the data is kept. <further details>*
- *The contract between the EEA customer and the HP EEA entity receiving the information must contain appropriate data protection clauses (as described in the company Customer Privacy Rulebook <link to section>)*

In order to generate output for the DP stage, PRN parameters (i.e. rule conditions) are compared with the deal inputs (by direct mapping in the DB) and if all the values match (i.e. are equal, or the value is a member of the group defined in the corresponding PRN parameter, or the PRN parameter for that value is not specified – see below) then the PRN is triggered. The output for the DP phase is not just the output (flag and reasons) corresponding to one PRN but the aggregation of the output of all the PRNs that match the deal conditions, with an overall flag value calculated from the aggregated individual values.

The PRN parameters are the same as those associated with the deal inputs from the UIs. New parameters can be added by the privacy experts, but in order to be considered they have to be calculated for the deals also. So, the application automatically adds the new parameter to the list of inputs and for all the deals created from that time on, the new values will asked.

Sometimes the users do not have all the information asked, or they will have it in a later stage, and this is why both requirements and deals allow an unknown value for some inputs: this value is represented by ‘n/a’. A PRN value of ‘n/a’ will match with all deal input values for the same condition. The reverse is not the case, however.

The rules described above are used also within the DD phase, to derive the PRNs that apply to a deal, using updated deal inputs. For each PRN there is a set of Activities and Controls Nodes (ACN) that are stored in the DB and contain the controls or activities that need to be taken to implement the requirement contained in the PRN. This mapping is known *a priori* and is created by the privacy experts. Hence, the list of ACN that apply to a particular deal can be deduced. The output for this phase is a list of privacy controls or activities to be taken together with information about whether those are the outsourcing company or the client’s responsibility (that is additional information associated with each ACN in the DB).

A limitation of our current KB representation is that we do not address *conflict resolution* between requirements and leave their resolution to the end user. We also do not provide an automated mechanism to select between alternative controls of which a subset might already suffice to achieve a particular control objective. Our experience with end users shows that they still perceive significant value in being alerted to all the relevant issues and possible solutions in the first place. This is because users – and less experienced users in particular – are not aware of the different requirements with which they need to concern themselves, and so appreciate having all this information. However, it would be useful to be better able to resolve conflicts and remove redundant controls, etc. Doing this requires mapping the *content* of actual rules, and not just their trigger conditions. We believe that the use of ontologies can solve these issues and the authors have initiated research into privacy ontologies for use in a future version of this tool. See [3] for more details on PM.

4 A Generalization – Accountability and AMT

HP Labs and the HP Privacy Office are collaborating on a next generation tool for use within HP businesses that uses a similar approach but is designed to be applicable not just to outsourcing but to many different business domains (for example, marketing, product design etc). As some background, the privacy community recognizes a number of current privacy challenges including that globalization and new technologies are straining traditional frameworks for privacy. Furthermore, we are seeing the biggest change in privacy since the 1980s and there is uncertainty in all regions. This is a current theme in regulatory discussions, and *organizational accountability* is seen as an important part of the solution. Accountability was already mentioned in most of the original privacy guidelines (see for example [4]), and also forms part of many more current privacy frameworks: notably, Binding Corporate Rules (BCR) in EU and Crossborder Privacy Rules (CBPR) in APEC.

A high level definition of accountability is that it is “the obligation to act as a responsible steward of the personal information of others, to take responsibility for the protection and appropriate use of that information beyond mere legal requirements, and to be accountable for any misuse of that information” [5]. Weitzner *et al* define a related notion of ‘information accountability’ as meaning that “information usage should be transparent so it is possible to determine whether a use is appropriate under a given set of rules and that the system enables individuals and institutions to be held accountable for misuse.” [6]. They argue that a shift is needed for technology and policy from hiding information to ensuring that only appropriate uses occur. Technological means centering on policy-aware transaction logs, a language framework and reasoning tools underpin this approach.

Our AMT tool supports enterprise accountability: it helps an organization to ensure privacy concerns are properly and proactively taken into account in decision making in the businesses as well as provide some assurance that this is case. AMT analyses projects’ degree of compliance with HP privacy policy, ethics and global legislation, and integrates privacy risk assessment, education and oversight. The tool supplies individuals who handle data with sufficient information and guidance to ensure that they design their project in compliance. When an HP decision maker uses the AMT, they are initially taken through a series of customized questions and, based on their answers, a compliance report is automatically generated, a record is retained in the database and, if appropriate, HP Privacy team is notified. Where an issue has been identified, guidance is offered online that links into the HP Privacy Rulebook and checklists and reminders are provided. The user can use the tool in an educational ‘self-help’ mode, where his/her input is not logged. The report scores the project for a list of privacy risk indicators. In addition to this user perspective, the system provides a privacy team perspective which is a knowledge management interface for KB update and modelling of high level principles. This allows management dashboard views of privacy processes across HP to identify risks and trends and assure compliance. AMT uses a rules engine both for the customized generation of the questions that a user needs to answer as well as for the generation of an output report for the user. (The KB representation we have chosen in the BPO tool does not allow for such customization.) For further details about this system and its formal properties, see [7].

5 Comparison with Related Work

Policy specification, modelling and verification tools include EPAL [8], OASIS XACML [9], W3C P3P [10] and Ponder [11]. These policies, however, are at a different level to the ones we are dealing with in this paper, as for example they deal with operational policies, access control constraints, etc. and not a representation of country or context-specific privacy requirements. In addition they are targeted towards machine execution and the question of intermediate, human-actionable representation of policies has so far not been paid attention to in the policy research community. Related technologies in the Sparcle [12] and REALM projects [13] do not produce output useful for humans. OASIS LegalXML [14] has worked on creation and management of contract documents and terms, but this converts legal documents into an XML format that is too long to be human readable and not at the right level for the representation we need in our system. Breaux and Antón [15] have also carried out some work on how to extract privacy rules and regulations from natural language text. This type of work has a different focus then ours but could potentially be complementary in helping to populate the KB more easily.

The translation from privacy laws to human-readable policies to machine-readable policies cannot be an exact one. Translation of legislation/regulation to machine readable policies has proven very difficult, although there are some examples of how translations of principles into machine readable policies can be done, e.g. PISA project [16], P3P [10] and PRIME project [17] .

The tool we have built is a type of expert system, as problem expertise is encoded in the data structures rather than the programs and the inference rules are authored by a domain expert. Techniques for building expert systems are well known [18]. A key advantage of this approach is that it is easier for the expert to understand or modify statements relating to their expertise. We are able to use a relatively simple underlying representation, as it was not necessary to use confidences, nor to schedule many rules that are eligible for execution at the same time through the use of a ‘conflict resolution’ strategy, as a one-step reasoning process sufficed.

Our systems can also be viewed as decision support systems (DSS). Many different DSS generator products are available, including [19,20]. All use decision trees or decision tables which is not suitable for our use as the reasoning in our system needs to be more complex than a simple decision tree can conveniently express.

In summary, our research differs from preceding research in that we define an intermediate layer of policy representation that reflects privacy principles linked into an interpretation of legislation and corporate policies and that is human-actionable and allows triggering of customised privacy advice.

6 Conclusions

There is a need for privacy technology to address business processes directly, and this paper has presented an approach to this problem that has a wide applicability. Specifically, we have developed, tested and deployed a privacy decision support tool (i.e. PM) within a corporate environment. The tool gathers user context for selected business processes and uses this to produce targeted privacy advice. It is being used on an ongoing basis in order to provide dynamic intra-company privacy advice in relation to a specific business process. In addition we have briefly introduced accountability as a promising concept for effective privacy protection and have built a prototype that enables accountable privacy management in large organizations. We believe that such tools will enable cutting-edge and effective privacy management for global organizations. Improving and creating further technologies for accountability in privacy management poses an interesting technical challenge for future research.

Acknowledgments. We would like to thank Prasenjit Dey from HP Labs India, Rahul Biswari and Prashant Choudhary from HP BPO, Andreia Esteves, Gareth Gale and Allan Parry from HP Labs Bristol, Allan Paull, Pierre Denereaz and Ray Farraro from the HP Privacy Office and Satish Patruni from HP Global Development India Center (GDIC) for their numerous contributions to this project.

References

1. Hecker, M., Dillon, T.S., Chang, E.: Internet Computing Privacy Ontology Support for E-Commerce, vol. 12(2), pp. 54–61. IEEE Computer Society Press, Los Alamitos (2008)
2. Martimiano, L.A.F., Goncalves, M.R.P., dos Santos Moreira, E.: An ontology for privacy policy management in ubiquitous environments, NOMS, pp. 947–950. IEEE, Los Alamitos (2008)
3. Pearson, Sander, Sharma. Privacy Management for Global Organizations, HP-TR (2009)
4. Organization for Economic Co-operation and Development (OECD): Guidelines Governing the Protection of Privacy and Transborder Flow of Personal Data, OECD, Geneva (1980)
5. Galway Project, Plenary Session Introduction, p. 5 (April 8, 2009)
6. Weitzner, A., Berners-Lee, F., Hendler, S.: Information Accountability. Communications of ACM 51(6) (June 2008)
7. Pearson, S., Rao, P., Sander, T., Parry, A., Paull, A., Patruni, S., Dandamudi-Ratnakar, V., Sharma, P.: Scalable, Accountable Privacy Management for Large Organizations. In: INSPEC 2009. IEEE, Los Alamitos (2009)
8. IBM: The Enterprise Privacy Authorization Language (EPAL), EPAL specification, v1.2 (2004), `http://www.zurich.ibm.com/security/enterprise-privacy/epal/`
9. OASIS: eXtensible Access Control Markup Language (XACML), `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml`
10. Cranor, L.: Web Privacy with P3P. O'Reilly & Associates, Sebastopol (2002)
11. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification Language (2001), `http://www-dse.doc.ic.ac.uk/research/policies/index.shtml`

12. IBM: Sparcle project,
`http://domino.research.ibm.com/comm/research_projects.nsf/pages/sparcle.index.html`
13. IBM: REALM project,
`http://www.zurich.ibm.com/security/publications/2006/REALM-at-IRIS2006-20060217.pdf`
14. OASIS: eContracts Specification v1.0 (2007),
`http://www.oasis-open.org/apps/org/workgroup/legalxml-econtracts`
15. Travis, D., Breaux, T.D., Antón, A.I.: Analyzing Regulatory Rules for Privacy and Security Requirements. IEEE Transactions on Software Engineering 34(1), 5–20 (2008)
16. Kenny, S., Borking, J.: The Value of Privacy Engineering, JILT (2002)
17. Privacy and Identity Management for Europe (2008),
`http://www.prime-project.org.eu`
18. Russel, S., Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice-Hall, Englewood Cliffs (2003)
19. Dicodess: Open Source Model-Driven DSS Generator,
`http://dicodess.sourceforge.net`
20. XpertRule: Knowledge Builder,
`http://www.xpertrule.com/pages/info_kb.htm`

Obligation Language and Framework to Enable Privacy-Aware SOA

Muhammad Ali, Laurent Bussard, and Ulrich Pinsdorf

European Microsoft Innovation Center
Ritterstr. 23, 52072 Aachen, Germany
{lbussard,ulrich.pinsdorf}@microsoft.com,
muhammad.ali@rwth-aachen.de

Abstract. Privacy policies defines rights and obligations on data (e.g. personally identifiable information) collected by services. Tackling privacy policies in a service oriented architecture spanning multiple trust domains is difficult because it requires a common specification and distributed enforcement. This paper focuses on the specification and enforcement of obligations. We describe the requirements, the resulting language, and its implementation. Finally, we compare our results with obligation support in the state of the art. The key contribution of this work is to bridge the gap between specific mechanisms to enforce obligations and underspecified support for obligations in today's access control and data handling policy languages.

1 Introduction

Data handling is an important part of privacy that reaches beyond pure access control. While access control defines *whether* access to data is granted, data handling policies define *how* data has to be handled after the access was granted. Data handling consists of two parts: first, it defines rights of the data consumer to store, process, and share a given piece of data. Second, it defines obligations that the data consumer has to commit to. In this paper we focus on the obligation part of data handling. Enforcing obligations is very challenging in service oriented architectures (SOA), where data can be collected by composite services (e.g. Mash-up) spanning multiple trust domains. To handle obligations in SOA, we assume that the framework described in this paper is deployed in each trust domain. In this paper, we define an *obligation* as:

> "A *promise* made by a *subject*[1], to a *user*. The subject is expected to fulfill the promise by executing and/or preventing a specific *action* after a *particular event*, e.g. time, and optionally under certain *conditions*".

Obligations play an important role in daily business. Most companies have a process to collect personally identifiable information (PII) on customers and ad-hoc mechanisms to keep track of associated rights and obligations. State of

[1] In obligation terminology, the subject is the subject of the obligation, i.e. the service. Do not confuse with the "data subject", which is the user in privacy terminology.

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 18–32, 2010.

the art mechanisms to handle collected PII accordingly to a privacy policy are lacking expressiveness and/or support for cross-domain definition of obligations. Please refer to Sect. 5 for a complete evaluation of the state of the art.

We identify four main challenges related to obligations.

1. Service providers must avoid committing to obligations that cannot be enforced. For instance, it is not straightforward to delete data when backup copies do exist. Tools to detect inconsistencies are necessary.
2. Services should offer a way to take user's preferences into account. Preferences may be expressed by ticking check boxes, be a full policy, or even be provided by a trusted third party. Mechanisms to match user's privacy preferences and service's privacy policies are necessary.
3. Services need a way to communicate acceptable obligations to users, link obligations and PII, and enforce obligations.
4. Finally, users need a way to evaluate the trustworthiness of service providers, i.e. know whether the obligation will indeed be enforced. This could be achieved by assuming that misbehavior impacts reputation, by audit and certification mechanisms, and/or by relying on trusted computing.

While our overall research addresses aspects of those four topics, the contributions of this paper are mainly related to the third aspect. Section 3 describes our proposed general-purpose obligation language. It is expressive enough to specify complex real-world obligations and can be extended with domain specific triggers and actions. This obligation language enables cross-domain scenarios where obligations must be semantically understood on user-side and service-side. It also enables both data handling policies and access control policies to make practical use of obligations. Section 4 gives an overview on the architecture for exchanging and enforcing obligations. Section 5 compares our work with the state of the art, while Sect. 6 summarizes the paper and provides and outlook on future work.

2 Requirements for Obligations

This section describes general requirements for an obligation language and enforcement framework. This list was mainly compiled by looking at scenarios we address in the PrimeLife project[2] and requirements found in related work [1,2,3,4,5].

Requirement 1: *Independence from policy language.* Obligations can be enforced independently from the embedding policy languages offering the placeholder for the obligation. Thus the obligation framework should be able to enforce obligations defined by the service, e.g. XACML [4] or P3P [6], and obligations in sticky policies, e.g. PRIME-DHP [1].

Requirement 2: *Independence from data storage.* The obligation handling must be independent from the concrete data store. The obligation travels with the data and should be stored along with the data so that the reference does not get lost.

[2] `http://primelife.eu/`

For instance, the obligation may refer to personal data stored in a database or to documents in a file system. Moreover, one obligation may refer to multiple pieces of data.

Requirement 3: *Independence from communication protocols.* The framework must be independent from the communication protocol. For instance, Web Services, REST, or plain HTTP could be used to exchange data and obligations.

Requirement 4: *Support for common obligations.* The obligation language should be extensible but not empty. Usual actions such as, for instance, *delete*, *anonymize*, *notify user*, *get approval from user*, *log* should be available with different implementations. It should support time-based and event-based triggers.

Requirement 5: *Support for domain specific obligations.* The framework must be open to define additional domain specific obligations. This requires mechanisms to define new types of actions and triggers. Naturally, the semantics of these new elements has to be understood by all stakeholders.

Requirement 6: *Support for abstraction of actions.* The obligation language must offer abstract actions which are configurable for the specific purpose. For instance, a *notify user* action might be implemented as sending an e-mail, sending an SMS, sending a voice message, or calling (and authenticating to) a web service.

Requirement 7: *Support for abstraction of triggers.* The obligation language must offer abstract and configurable triggers. For instance, a trigger "access PII" may react both to a query on a database and to a read operation on a file server.

Requirement 8: *Support for distributed deployment.* Various deployments of the obligation framework can be envisioned: a corporate-wide obligation framework could cover multiple databases, a desktop obligation framework could deal with local files, or it could even be provided as a "cloud service". In any case, only one obligation framework has to be in charge of a specific piece of data.

Requirement 9: *Support for different trust models.* Users have to trust the service provider, i.e. assume that it will fulfill obligations. The anchor of trust could be based on various technologies, e.g. a trusted stack (certified TPM [7], trusted OS), reputation, or certification by external auditors. The obligation language should be independent from the trust model.

Requirement 10: *Transparency of data handling.* The obligation enforcement as well as mechanisms to load policies should be comprehensive so that data processors and auditors can easily check whether a specific deployment is compliant with a given specification. This is a prerequisite to enable data-handling transparency toward end users.

Requirement 11: *Support for preventive obligations.* The obligation language should be able to express preventive statements that forbid the execution of an action. For instance, the obligation to store logs for six months forbids the deletion of log files.

3 Language for Obligations Description

Formally, we represent an obligation o as a tuple $\langle s, a, \xi, c, e \rangle$ where s is a subject which is obliged to fulfill the obligation, a is an action which is executed/

prevented to fulfill the obligation, ξ is a set of triggers, c is a boolean equation specifying conditions under which the obligation rule would be active and e is the set of events which are sent outward in case of a change in state of obligation e.g. violation or fulfillment. We use O as the set of all possible obligations.

Subject. Subject s is an identifier for the data processing party that needs to obey an obligation; $s \in S$ where S denotes the set of all existing subject identifiers.

Action. Action a is the activity executed to fulfill an obligation and is represented as a tuple $\langle i, p, at \rangle$ with $i \in I$, where I represents the set of all possible action identifiers. Each element of I can be uniquely mapped to available actions within the system using a bijection $map : I \rightarrow A$. The action parameters p is a set of name/value pairs. We classify actions by their action type $at \in \{proactive, preventive\}$

- *Proactive* actions which require the execution of actions pro-actively. For example Delete, SendEmail etc.
- *Preventive* actions which can only be prevented from executing to fulfill the corresponding obligation promise. This class of actions add lot of expressiveness into our language and does allow negative obligation statements like *Subject X commits never Sharing U's data with anyone* where the action *never share* itself is never executed but the fulfillment is done by preventing the action *share*.

We do not allow actions which can be used as both proactive and preventive within the system.

An obligation rule contains a single action, however we envision that this action itself can be composed of many basic actions arranged in a complex manner. This restriction has been put to avoid ambiguity. Indeed, if we would have two actions, e.g. *Delete* and *SendNotification*, in the same rule and the first one is executed successfully while the second fails the overall status of the rule is undecidable (fulfilled or violated). We consider deciding the status of rules having multiple actions as a difficult problem which is part of our future work.

Triggers. Triggers define the types of inward events which result in the execution of the obligation's action. Multiple triggers can be defined for a single obligation. Triggers can be *deterministic* where we know the precise time when the trigger will be fired and we classify them as *AbsoluteTimeTriggers*. Such triggers can be defined by a tuple $\langle \tau, d \rangle$ where τ is any absolute point in future time and d is the timeout duration. The rule must be fulfilled before $\tau + d$. Deterministic triggers, in conjunction with temporal conditions, provides the capability to express time bounded obligations as the rule activation time frame and time to trigger execution are known in advance.

Trigger can also be *non-deterministic* and fired in reaction to event locally or externally generated. For performance reasons, we suggest that these events should contain the user data unique identifier that is used to select the corresponding policy and in turn related obligation. Beside this required information external triggers may accompany additional parameters depending on their type. Non-deterministic trigger is defined by a tuple $\langle ty, d \rangle$ where $ty \in T$, where T denotes the set of all existing trigger types in the system and d is the deadline duration as defined before. Parameters for trigger are not specified within the policy but the triggers when fired may specify them.

Application Condition. Application condition expressions are boolean equations defining whether a rule is applicable. When an event occurs (e.g. delete resource r), the system takes into account any obligation $\langle s, a, \xi, c, e \rangle$ having triggers ξ registered to such events. If the condition c of an obligation is evaluated to true, the obligation's action is executed.

Depending on the result of the action, the obligation will be considered as fulfilled or violated. If the obligation is non-repeating, it will disappear from the system after fulfillment or violation.

The condition is expressed as sum of products. We have shown the grammar of condition expression (cexpr) in EBNF form below.

$$\begin{aligned}
cexpr &= \{pterm\};\\
pterm &= \{cond|cexpr\};\\
cond &= name, \{parameter\};\\
parameter &= function|variable|literal;\\
function &= returntype, name, \{parameter\};\\
variable &= name, type\\
literal &= \{a...z|A...Z|0...9\}
\end{aligned}$$

For example, in order to have temporal constraints on the obligation rule we can define a time frame function which can then be used in policies. We give an example condition below:

$$\begin{aligned}
cexpr = (Timeframe(t_s, t_e) \wedge UsageLimit(i)) \vee \\
(System.State == green)
\end{aligned}$$

Timeframe and *UsageLimit* are the function/condition names. Conditions are subset of functions which always return a *boolean* and thus can be used in the product terms (pterm). The second product term specifies that if the environment variable *System.State* is *green* then condition is applicable. The environment variables are specified in the *variable repository* (see Section 4). The two product terms are OR-ed together.

Events. Optionally obligation rules can have outward events which are generated when a state of the rule is changed. Any event resulting from an obligation rule can be a trigger for another rule within the same policy. Through this design we implement the notion of cascaded obligations. Formally, $e \in E$ where E denotes the set of existing events.

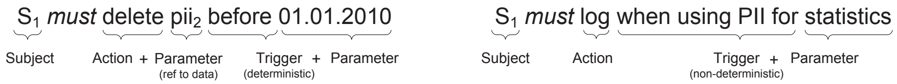


Fig. 1. Example of obligation rules

Examples. Figure 1 shows two basic examples of obligation rules in pseudo language. The first rule is triggered by time and leads to the deletion of a specific piece of data. The second rule is triggered by actions on a piece of data and results in logs.

3.1 Obligation Policy

An obligation policy is a set of well defined obligation rules which are consistent as a whole. Formally

$$\rho = \{o : o \in O\} \wedge |\rho| > 0$$

It is of prime importance that the policy must be consistent. Inconsistencies like self-contradiction, infinite cascading through cyclic dependencies must be absent from a policy.

Let $o_1 = \langle s_1, a_1, \xi_1, c_1, e_1 \rangle$ and $o_2 = \langle s_2, a_2, \xi_2, c_2, e_2 \rangle$ with $o_1, o_2 \in \rho$ be two obligations, where ρ is an obligation policy as defined before. We use the operator $\bowtie$ to represent *semantic contradiction* between two entities. This is a symmetric, non-reflexive and non-transitive relation.

An obligation policy ρ is inconsistent if one of the following conditions is true:

1. ρ has an obligation rule which is inconsistent, i.e. $\exists o \in \rho : o\, \textit{is not enforceable}$.
2. ρ contains two semantically contradicting obligation rules, i.e. $\exists o_i, o_j \in \rho : o_i \bowtie o_j$.

The first case arises when any rule within ρ is not enforceable because it contains actions or conditions whose processing plug-ins are not present within the system. Policies must be validated before being deployed in the templates repository.

The second case occurs when two rules, having the same subject and overlapping conditions, are contradicting each other because of contradicting actions. If the condition are not overlapping then it is not necessarily a consistency error. We define function $IsConditionOverlap(c_1, c_2) \in \{true, false, undefined\}$ which establish whether the two rules could become active within the same time frame in future. If they do then they may be triggered both at the same time and because of the contradiction it would be impossible to execute both actions. At design time of an action plug-in, we define explicitly which actions are contradicting others within the system. This meta information aid policy writers to write consistent policies.

$$\text{If } a_1 \bowtie a_2 \ \wedge \ s_1 = s_2 \wedge \ IsConditionOverlap(c_1, c_2) = true \Rightarrow o_1 \bowtie o_2$$

If the condition overlap relation is not decidable we can only raise a warning to the policy writer. We could take *undefined* as a consistency error, but that will reduce the expressiveness of the language.

$$\text{If } a_1 \bowtie a_2 \wedge s_1 = s_2 \wedge IsConditionOverlap(c_1, c_2) = undefined \Rightarrow o_1 \bowtie o_2 \text{ is undefined}$$

Otherwise, *IsConditionOverlap* returns *false* which ensures that the two rules, having contradicting actions, would be active in a separate time frames and it is safe to have them within the policy. There is also a possibility of having action precedence with some actions which cannot be repeated e.g. *Delete User Data* which is non-repeatable action as once the data is deleted it cannot be deleted again. Similarly, after the deletion of data the existence of policy itself, attached to deleted data, may vanish so the obligation rules which are supposed to be executed after *delete* action may become *redundant* or *non-reachable.* The current implementation does not yet target such complex cases.

Infinite cascading of rules because of the presence of events which can trigger other rules within the same policy is also a problem. It must be ensured that infinite cascading of rules must not happen and cycles are identified at policy writing time.

3.2 Additional Aspects of Obligations

Obligation rules could be subject to certain generic and temporal conditions which are prerequisite to obligation rule fulfillment. This key aspect is addressed by having an *application condition* construct in our proposed obligation rule. Conditions could even be stateful for instance take the statement *subject X commits to Send Account Statement three times a year* where the state of the condition should be tracked to establish the rule applicability.

In case of temporal conditions the time frame in which the rule would be active is specified explicitly which makes the rule as time bounded. Alternatively, we can have only non-temporal conditions but their fulfillment is non deterministic. In the absence of temporal conditions the obligation rule can be time unbounded.

Cyclic or repeating obligations are required to be fulfilled multiple times. This aspect has been incorporated by allowing multiple triggers to be defined for a single obligation rule.

There are some aspects which are not addressed until now but worth mentioning here. We consider that obligation subject could be more complex than just an identity. We could have an individual entity who has the full responsibility of fulfilling obligations or collection of entities forming a logical subject and the responsibility division to fulfill the obligation in turn could be complex like *All*, *one out of all* etc. In real world we could even have one entity committing something on behalf of another based on some underlying reason e.g *Authority*, *Mutual agreement* etc. *Observability* or *monitoring* of obligation fulfillment is another important aspect as also being discussed in [5]. We do consider that monitoring of obligations is an attribute of the rule as well as dependent on the reference monitor scope whether deployed within the same trust domain or outside. In the current work we have not addressed these problems.

4 Architecture for Enforcement

We have designed and implemented an enforcement architecture for the obligations which are expressed through our proposed language. The core requirements of the architecture were to ensure the enforcement of obligations, to enable customized actions, to facilitate integration with existing systems and to support external systems. The detailed obligation framework architecture is illustrated in Figure 2.

The key feature of the framework is its flexibility which is achieved through the plug-in based design allowing easy integration of new types of obligations and new types of external systems. The framework uses the available plug-ins to execute different tasks. We assume that the framework is authorized to perform all the obligation actions on the external entities (e.g. databases, email servers).

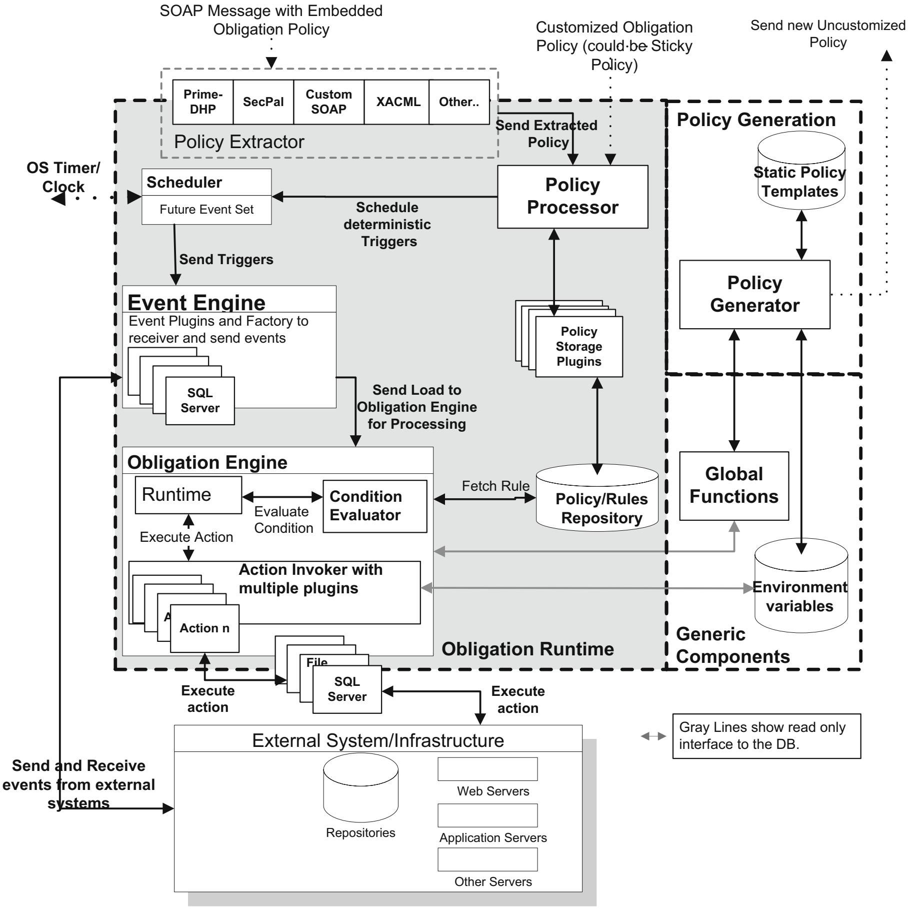


Fig. 2. Obligation Framework Architecture

This is generally achieved by deploying obligation framework and external systems within one single trust domain. As shown in Figure 2, the architecture is separated into three main parts, namely *Policy generation*, *Generic components*, and *Obligation Runtime*.

Policy Generation Components are used mainly for policy creation. The underlying idea is to store obligation policies in the form of *policy templates* with annotated fields. Once the request is received for a new policy, to be sent to the user, one of those templates is extracted from the repository based on the context of the request and is sent back to the user.

Generic Components are also an optional set of components used to store environmental variables, global functions etc.

Obligation Runtime Components are the core components within the architecture. We now discuss each of the subcomponents of the obligation runtime briefly.

Policy Extractor plug-ins. Those plug-ins extract the obligation expression from the incoming message which could be in any format, as long as the required translation/extraction plug-in is present. If the obligation policy is embedded within any other container message, the corresponding plug-in parses the message and forwards only the obligation policy part to the system. This enables fulfillment of requirements 1, 2 and 3.

Policy Processor. The policy is received by the *policy processor* either through an external interface or via any of the *policy extractor plug-ins*. It processes the policy, check inconsistencies, and schedule deterministic triggers. The initial transaction interplay with the user ends here and the system returns the system wide unique *Policy ID* to the caller which forwards it to the user.

The caller in turn stores PII somewhere within the infrastructure of subject along with the policy reference. Both data and policy are stored separately but remain connected through cross-references. Thus, the enforcement framework only manages policy templates, policies connected to some data under the subject ownership and references to that data. Data itself is being managed by systems external to our obligation framework, but within the subject's trust domain.

Scheduler. The scheduler is used to initiate time based triggers which are scheduled by other components of the runtime engine. The triggers are being in the form of messages to the event engine. We refer to the scheduled triggers as *future event set*.

Event Engine. This is the central collection and distribution component. The major goal to have a single point of event receiving and distribution is to ensure integrity. All the external systems, scheduler and obligation engine communicate to other components through the event engine. This component also keeps track of the received and processed messages. Storing and retrieving these active messages in case of system shut down or malfunction is also the responsibility of this central component. It behaves mainly like a queuing component. This design allows us to integrate our framework with existing systems and to enforce preventive obligations which are fulfilled by inhibiting rather performing an action which is our requirement 11.

Obligations Engine. The obligation engine is the main load processing component. It received the load/triggers from the event engine and processes them. On receiving a new trigger, it fetches the policy rules from the policy repository, evaluates conditional statements, finds the respective action plug-in and executes the action. After the execution of actions, the obligation engine changes the state of obligation rule and fires the outward events. If the action is executed successfully before the deadline the rule is fulfilled otherwise violated.

The policies contain *action* with parameters attached to each obligation rule. Each of these actions must match to an available *action plug-in* within the obligation engine. The parameters listed within the policy must also match to the parameters required by the actions plug-in. This enables fulfillment of requirements 5 and 6.

In the obligation engine component, we propose a two-layer action plug-in mechanism. The upper layer contains the plug-ins for specific actions e.g *delete*, *notify* and the lower plug-in layer contains the implementations for different external systems supporting a set of actions. For instance, delete operation can operate on files or on data in a relational database. Notification to user could be sent via e-mail, fax, or postal mail. Each obligation policy rule in our language specifies a single action with a system-wide unique scope and name, which is used to select appropriated plug-in.

To ensure integrity, the action parameters must satisfy the required parameters for only one lower layer plug-in. This design ensures the requirements 4, 6 and 7.

We kept our language independent of schema extensions so the new vocabulary required for domain specific obligations is mainly added by implementing the corresponding plug-ins each having unique scope and name which are then used within the policy. This targets our requirement 8 for the enforcement platform. Requirement 9 on trust model and 10 on transparency are not yet covered and need additional research.

5 Comparison with State of the Art

Most of the available policy languages, like XACML [4,8], EPAL [9], Ponder [10], Rei [11] and PRIME-DHP [1], provide either only a placeholder or very limited obligation capability. Moreover these languages do not provide any concrete model for obligation specification. XACML and EPAL support system obligations only, as no other subject can be expressed in their proposed language. Ponder and Rei on the other hand do allow user obligations, however they do not provide a placeholder explicitly for the specification of temporal constraints and they do not support pre-obligations, conditional obligations, and repeating obligations.

PRIME-DHP proposed a new type of policy language which expresses policies as a collection of data handling rules which are defined through a tuple of recipient, action, purpose and conditions. Each rule specifies who can use data, for what purposes and which action can be performed on the data. The language structure is flat which limits its expressiveness. PRIME-DHP itself also does not provide any concrete obligation model.

Besides the policy languages, we observed publications on expression, enforcement and formalization of obligations. In the next paragraphs, we collected prior art which is directly related to our approach and point out the key differences to our work.

Mont Casassa et al. [2] proposed the idea of having parametric obligation policies with actions and events having variable parameters. This work was done in conjunction with the PRIME-DHP to support obligations. It is by far the closest work to ours. They propose a formal obligation model and provide the framework to enforce obligations. However, they do not offer the notion of preventive obligations (negative obligations) and multiple subjects. As opposed to their policy expressions, we propose a schema which is not modified when domain specific obligations, including new actions, events and triggers, are added. They took the notion of *On violation actions* within a policy rule to express actions which are taken in case of obligation violation. We cover this aspect by defining that obligations rules contain *events* which can be used to trigger another rule within the same policy to invoke a compensatory action. Since this event-based approach allows cascading of rules, we need to ensure the absence of loops, which remains an open issue of our work. Unlike [2], we also do not allow multiple action per rule because of the system integrity problem which arises from the fact that we cannot map fulfillment of a subset of actions in any policy rule as complete fulfillment and we achieve the same behavior through rule cascading without ambiguity.

Irwin et al. [3] proposed a formal model for obligations and define secure states of a system in the presence of obligations. Furthermore, they focused on evaluating the complexity of checking whether a state is secure. However, the proposed obligation model is very restricted and neither support pre-obligations (provisions) nor repeating and conditional obligations, which are required in different domains and scenarios. They addressed the problem of verification of obligation enforcement while we focus on the expression of a wide range of scenarios, supporting all of the above types of obligations. In other words, the two research efforts are targeting different problems.

Pretschner et al. [5, 12, 13] worked in the area of distributed usage control. In [5], they used distributed temporal logics to define a formal model for data protection policies. They differentiated provisional and obligation formulas using temporal operators. Provisions are expressed as formulas which do not contain any future time temporal operators and obligation are formulas having no past time temporal operators. They also addressed the problem of observability of obligations which implies the existence of evidence/proof that the reference monitor is informed about the fulfillment of obligations. Possible ways of transforming non-observable obligations into observable counterparts have also been discussed. We also consider temporal constraints as an important part of obligation statement. However, we deem observability as an attribute of the reference monitor and not an attribute of the obligation rule. It depends on the scope of the monitor. The scope could be within the system, within the same trust domain but outside the system, or even sitting outside the trust domain, to observe fulfillment and violations. We currently have not addressed this problem of observability. In [12] they have proposed an obligation specification language (OSL) for usage control and presented the translation schemes between OSL and rights expressions languages, e.g. XrML, so the OSL expression could be enforced using DRM enforcement mechanisms. We have tried to fill that gap by implementing the enforcement platform for enforcing obligation policies without translation. In [13], the authors have addressed the scenario of policy evolution when the user data crosses multiple trust domains and the sticky policy evolves. Currently, we are not focusing on evolution of obligation policy, but it could likely be one of the future extensions of our work where we plan to address the interaction of obligation frameworks at multiple services which is complementary to what is discussed in [13].

Katt et al. [14] proposed an extended usage control (UCON) model with obligations and gave a prototype architecture. They have classified obligations in two dimension a) system or subject performed and b) controllable or non-controllable where the objects in the obligation would be either controllable or not. Controllable objects are those that are within a target systems domain, while non-controllable objects are outside the systems domain. The enforcement check would not be applied for system-controllable obligations where they assume that since system is a trusted entity so there is no need to check for the fulfillment. The model does not address the conditional obligations.

Rakaiby et al. [15] as well as Cholvy et al. [16] studied the relationship between collective and individual obligations. As opposed to individual obligations which are rather simple as the whole responsibility lies on the subject, collective obligations are targeted toward a group of entities and each member may or may not be responsible to fulfill those obligations. We also consider that the subject of any obligation rule is a complex entity in itself like individual or group, self directed or third party. Our current implementation does not support this but could be extended to include such scenarios.

Ni et al. [17] proposed a concrete obligation model which is an extension of P-RBAC [18]. They investigated a different problem of the undesirable interactions between permissions and obligations. The subject is required to perform an obligation but does not have the permissions to do so, or permission conditions are inconsistent with the obligation conditions. They have also proposed two algorithms, one for minimizing invalid permissions and another for comparing the dominance of two obligations. Dominance relation is the relationship between two obligations which implies that fulfillment of one obligation would cover the fulfillment of other which is analogous to set containment.

Gama et al. [19] presented an obligation policy platform named *Heimdall* which supports the definition and enforcement as a middleware platform residing below the runtime system layer (JVM, .NET CLR) and enforcing obligations independent of application. Opposed to that, we present an obligation framework as an application layer platform in a distributed service-oriented environment which could be used as an standalone business application to cater for user privacy needs. We believe that it is not necessary to have the obligation engine, which is an important infrastructure component to ensure compliant business processes, as part of the middleware. Moreover our service-oriented approach supports interoperability in an heterogeneous system environment.

The work present in this paper incorporates some of the prior art and extends it toward more expressiveness, extendability, and interoperability. However, we think that some authors addressed different problems, and it would be worthwhile to further combine their results with our approach.

6 Conclusion and Future Work

This paper described challenges and requirements to properly address obligations. We presented a general language for obligations, which can be used with today's access control and data handling policy languages. The language offers basic actions, triggers, and terms that are rich enough to cover a broad range of scenarios. In addition the language can be extended with domain specific actions and triggers to adapt it to specific application domains.

The reasoning is based on an abstract yet expressive obligation language. We presented eleven requirements for design and an abstract notion of an obligation language fulfilling them. We described important design aspects and formal structure of the obligation language. We verified our work with an implementation of an obligation framework which features both the requirements and the proposed language design. This allowed us to make practical comments on the implementation aspects. Finally we showed how our work relates to the state of the art.

Future work will be aligned with challenges described in the introduction of this paper. First we need mechanisms to help authors of privacy policies to check whether a policy can be enforced. This is especially important since violation of obligations impacts reputation and can have legal implications. Next, we need to look at the protocols and matching mechanisms between users (with privacy preferences) and service providers (with privacy policies). Both issues require specifying the semantic of obligations, i.e. the semantic of triggers and actions, in order to compare them. Finally, we will also consider distributed services where collected PII is subsequently shared with third parties. From an obligation perspective, the key interest is to look at distributed yet coherent enforcement.

Acknowledgment

The research leading to the results presented in this paper has received funding from the European Community's Seventh Framework Program (FP7/2007-2013)

for PrimeLife project. The information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The PrimeLife consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

References

1. Ardagna, C.A., Cremonini, M., De Capitani di Vimercati, S., Samarati, P.: A privacy-aware access control system. J. Comput. Secur. 16(4), 369–397 (2008)
2. Casassa, M., Beato, F.: On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises. In: Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, 2007, pp. 51–55. IEEE Computer Society Press, Los Alamitos (2007)
3. Irwin, K., Yu, T., Winsborough, W.H.: On the modeling and analysis of obligations. In: CCS 2006: Proceedings of the 13th ACM conference on Computer and communications security, pp. 134–143. ACM, New York (2006)
4. Rissanen, E.: OASIS eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS working draft 10, OASIS (March 2009)
5. Hilty, M., Basin, D., 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)
6. Cranor, L., Langheinrich, M., Marchiori, M., Reagle, J.: The platform for privacy preferences 1.0 (p3p1.0) specification. W3C Recommendation (April 2002)
7. TCG: Trusted Computing Platform Alliance (TCPA). Main Specification Version 1.1b, Trusted Computing Group, Inc. (February 2002)
8. Moses, T.: OASIS eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard oasis-access_control-xacml-2.0-core-spec-os, OASIS (February 2005)
9. IBM: Enterprise privacy authorization language (EPAL 1.2)
10. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Sloman, M., Lobo, J., Lupu, E.C. (eds.) POLICY 2001. LNCS, vol. 1995, pp. 18–38. Springer, Heidelberg (2001)
11. Kagal, L., Finin, T., Joshi, A.: A policy language for a pervasive computing environment. In: POLICY 2003: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks, p. 63. IEEE Computer Society, Los Alamitos (2003)
12. Hilty, M., Pretschner, A., Basin, D., Schaefer, C., Walter, T.: A policy language for distributed usage control. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 531–546. Springer, Heidelberg (2007)
13. Pretschner, A., Schütz, F., Schaefer, C., Walter, T.: Policy evolution in distributed usage control. In: 4th Intl. Workshop on Security and Trust Management, Elsevier, Amsterdam (2008)
14. Katt, B., Zhang, X., Breu, R., Hafner, M., Seifert, J.P.: A general obligation model and continuity: enhanced policy enforcement engine for usage control. In: SACMAT 2008: Proceedings of the 13th ACM symposium on Access control models and technologies, pp. 123–132. ACM, New York (2008)

15. El Rakaiby, Y., Cuppens, F., Cuppens-Boulahia, N.: Formalization and management of group obligations. In: Proceedings of IEEE International Symposium on Policies for Distributed Systems and Networks, POLICY 2009 (2009)
16. Cholvy, L., Garion, C.: Deriving individual obligations from collective obligations. In: AAMAS 2003: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pp. 962–963. ACM, New York (2003)
17. Ni, Q., Bertino, E., Lobo, J.: An obligation model bridging access control policies and privacy policies. In: SACMAT 2008: Proceedings of the 13th ACM symposium on Access control models and technologies, pp. 133–142. ACM, New York (2008)
18. Ni, Q., Trombetta, A., Bertino, E., Lobo, J.: Privacy-aware role based access control. In: SACMAT 2007: Proceedings of the 12th ACM symposium on Access control models and technologies, pp. 41–50. ACM, New York (2007)
19. Gama, P., Ferreira, P.: Obligation policies: An enforcement platform. In: POLICY 2005: Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks, Washington, DC, USA, pp. 203–212. IEEE Computer Society, Los Alamitos (2005)

Distributed Privacy-Preserving Methods for Statistical Disclosure Control

Javier Herranz[1], Jordi Nin[2], and Vicenç Torra[3]

[1] Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya,
C. Jordi Girona 1-3, Mòdul C-3, 08034 Barcelona, Catalonia, Spain
jherranz@ma4.upc.edu

[2] LAAS, Laboratoire d'Analyse et d'Architecture des Systèmes,
CNRS, Centre National de la Recherche Scientifique,
7, Avenue du Colonel Roche, 31077 Toulouse, France
jnin@laas.fr

[3] IIIA, Artificial Intelligence Research Institute,
CSIC, Spanish National Research Council,
Campus UAB s/n, 08193 Bellaterra, Catalonia, Spain
vtorra@iiia.csic.es

Abstract. Statistical disclosure control (SDC) methods aim to protect privacy of the confidential information included in some databases, for example by perturbing the non-confidential parts of the original databases. Such methods are commonly used by statistical agencies before publishing the perturbed data, which must ensure privacy at the same time as it preserves as much as possible the statistical information of the original data.

In this paper we consider the problem of designing distributed privacy-preserving versions of these perturbation methods: each part of the original database is owned by a different entity, and they want to jointly compute the perturbed version of the global database, without leaking any sensitive information on their individual parts of the original data. We show that some perturbation methods do not allow a private distributed extension, whereas other methods do. Among the methods that allow a distributed privacy-preserving version, we can list noise addition, resampling and a new protection method, *rank shuffling*, which is described and analyzed here for the first time.

Keywords: Statistical disclosure control, privacy, homomorphic encryption.

1 Introduction

There are many real situations where confidential data of people (respondents) is published by statistical agencies, to be used by decision makers, politicians, researchers, etc. This dissemination of confidential information should ensure, however, that the privacy of the respondents is protected in some way, to be in accordance with current laws and regulations [17]. For example, a person would not be happy if some published dataset contains a record with some attributes which identify him univocally, concatenated with some confidential attributes such as the income or the diseases he has suffered from.

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 33–47, 2010.

In this scenario, one approach to achieve some level of privacy is the application of *perturbative protection methods* to the confidential data, before making it public. There is a large number of such methods, based on noise addition [16], clustering [6], data swapping [4], resampling [14], and others. Besides protecting the privacy of the respondents, the main goal is that the protection method preserves as much as possible the statistical utility of the original data; for this reason, these methods are also known as *Statistical Disclosure Control* (SDC) methods. Clearly, but unfortunately, privacy and statistical utility go in opposite directions. A protection method is well considered if it offers a good trade-off between privacy and utility. Note that we consider a non-interactive scenario: the owner of the data applies some protection method to the whole database and publishes the resulting modified database. This is different to the interactive scenario where a client makes some queries on the (confidential) database, then the owner processes the queries and outputs some results. A lot of theoretical results for this latter scenario have appeared in the last years (see [12], for example).

In certain situations, it may be the case that the original data (that one wants to protect and release) comes from different servers or entities. They want to cooperate in order to produce a protected version of the whole database, but they do not want to leak any private information concerning their own individual parts of the original data. For example, many hospitals may want to cooperate with a research project dealing with a specific illness, by producing a global, but protected, database containing information on all (or some) of their patients. The released database will allow external researchers to compute some statistical values for their project, while privacy of the patients should be preserved by the application of a perturbative protection method. Furthermore, each hospital does not want anyone (including the rest of hospitals) to obtain any information on its individual part of the original data, only the information that can be derived from the final publication of the protected global database. For example, a hospital A does not want that other hospitals know how many patients in A suffer a certain illness, or how many patients are registered in A, or if there is some patient in A whose creatinine blood level is higher than any other patient of the rest of hospitals.

This paper deals with this problem of jointly and privately producing a perturbed database, where the original database is partitioned among several entities. The problem belongs to the general class of *multiparty computation* problems, for which there are general but inefficient solutions. The goal is to find more efficient solutions for this particular case where the function to be computed in a multiparty way is a perturbative protection function. Some works have considered this problem for the related case of clustering functions [1]; these results may be applied, in some cases, when the perturbative protection method is based on clustering techniques, e.g. microaggregation [6].

To the best of our knowledge, for other perturbation methods the problem is not trivial and has not been considered in the literature. In this work, we give both positive and negative results concerning distributed and privacy-preserving versions of these perturbation methods: random noise addition [16], rank swapping (and variants) [4,18], and resampling [14]. Namely, we first show that any distributed version of a method in the rank swapping family cannot offer a good enough level of privacy. To counteract this negative result, we introduce a new perturbative protection method, that we call *rank shuffling*. The design of this method follows ideas from rank swapping, but replacing the swapping step with a permutation (or shuffling) one. We have tested the new method to see that it obtains more or less the same quality results as other standard methods like rank swapping. Furthermore, this method allows a distributed and privacy-preserving version, that we describe in detail. This distributed protocol employs some well-known cryptographic primitives (although we propose some simpler variants of some of them), all based on homomorphic threshold cryptosystems. Our other results are also positive: we show that these same primitives can be used to design a secure multiparty version of noise addition and resampling.

This paper is organized as follows. In Section 2 we explain the concept of (threshold) homomorphic encryption, which will play a central role in the design of our distributed privacy-preserving protocols. We also describe the basic scenario where perturbative protection methods are applied to statistical databases, along with a brief explanation of some well-known perturbative methods. Then we propose in Section 3 a new perturbative method, rank shuffling, which intuitively (and experimentally) works better than the methods in the rank swapping family. Section 4 is entirely devoted to the study of distributed privacy-preserving versions of some perturbative protection methods. We conclude our work in Section 5.

2 Preliminaries

This section is devoted to the two main basic primitives that will be used in this paper: homomorphic encryption allowing re-randomization and threshold decryption, and perturbative methods for statistical disclosure control.

2.1 Homomorphic Encryption

A public key encryption scheme $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ consists of three probabilistic and polynomial time algorithms. The key generation algorithm $\mathcal{KG}$ takes as input a security parameter (for example, the desired length for the secret key) and outputs a pair (sk, pk) of secret and public keys. The encryption algorithm takes as input a plaintext m and a public key pk, along with some randomness, and outputs a ciphertext $c = \mathcal{E}_{pk}(m)$. Finally, the decryption algorithm takes as input a ciphertext and a secret key, and gives a plaintext $m = \mathcal{D}_{sk}(c)$ as output.

Such a scheme has an *homomorphic* property if there exist two operations, defined on the set of ciphertexts and plaintexts, respectively, such that the result of operating two ciphertexts is an encryption of the result of operating the two corresponding plaintexts. For example, a public key cryptosystem is *additively* homomorphic if there exists an operation $\oplus$ defined on the set of ciphertexts, such that the message encrypted in $c_1 \oplus c_2$ is $m_1 + m_2$, where m_i is the message encrypted in c_i, for $i = 1, 2$. Formally, this property is written as

$$\mathcal{D}_{sk}\big(\mathcal{E}_{pk}(m_1) \oplus \mathcal{E}_{pk}(m_2)\big) \;=\; m_1 + m_2.$$

Homomorphic cryptosystems have a lot of applications, including electronic auctions and electronic voting. Note that an additively homomorphic encryption scheme allows re-randomization: if $c = \mathcal{E}_{pk}(m)$ is an encryption of m, then $c' = c \oplus \mathcal{E}_{pk}(0)$ is a new and random encryption of m. We will need a cryptosystem that also supports secure (t,t)-threshold decryption: the key generation algorithm does not output sk but a share sk_i for each member of some set $\mathcal{P} = \{P_1, \ldots, P_t\}$ of t users; the encryption algorithm is the same, and the decryption algorithm must be jointly performed by all the t users in $\mathcal{P}$.

Paillier's cryptosystem [19] satisfies all the properties that we want: it is additively homomorphic and it supports (t,t)-threshold decryption, as shown in [13]. Therefore, we will use this cryptosystem as basis in the design of our distributed protocols, in Sections 4.4 and 4.5.

2.2 Statistical Disclosure Control (SDC)

A dataset X can be viewed as a matrix with n rows (*records*) and V columns (*attributes*), where each row contains V attributes of an individual. We denote by x_{ij} the value stored in row i and column j of X. The attributes in a dataset can be classified in two different categories, *identifiers* or *quasi-identifiers*, depending on their capability to identify unique individuals. Among the quasi-identifier attributes, we distinguish between *confidential* and *non-confidential*, depending on the kind of information that they contain.

We consider the following scenario for statistical disclosure control: (i) identifier attributes in X are either removed or encrypted, therefore we will write $X = X_{nc}||X_c$; (ii) confidential quasi-identifier attributes X_c are not modified; (iii) a *perturbative protection method* ρ is applied to non-confidential quasi-identifier attributes, in order to preserve the privacy of the individuals whose confidential data is being released. This leads to a protected dataset $X'_{nc} = \rho(X_{nc})$. This scenario, which was used first in [9], has also been adopted in many other works.

Besides protecting the privacy of the respondents, the main goal is that the data protection method preserves as much as possible the statistical utility of the original data. Of course, the values of privacy and statistical utility are inversely related. A particular perturbative protection method is well-considered if it achieves a good trade-off between privacy and statistical utility. There are different ways to measure this trade-off. Maybe the simplest and most intuitive one is the *score*, which was presented in [8]. It just measures the average between two quantities: one of them analyzes the information loss (IL) which is produced by the application of the protection method, and the other one counts the risk (DR) that an intruder can obtain any information that breaks the privacy of the data, after the protected dataset X' has been released. Since the score is a generic measure, it can be applied to any protection method ρ and is therefore a very good way to compare and classify different methods, as it was done in [9]. We briefly explain now some of the protection methods that appear in the ranking in [9], and that will be considered for the design of distributed privacy-preserving SDC methods.

Noise Addition. Noise addition is perhaps the simplest and most intuitive data perturbation method. To anonymize data by additive noise [16], each value x_{ij} of the original dataset X is replaced with $x'_{ij} = x_{ij} + \epsilon$, where ϵ is a normally distributed error drawn from a random variable $\epsilon \sim N(0, \sigma_\epsilon^2)$.

In the literature, the general assumption is that the variances of ϵ are proportional to those of the original attributes. Thus, if $\sigma^2_{at_j}$ is the variance of at_j, then $\sigma^2_\epsilon = \alpha\sigma^2_{at_j}$, for some α. In this way, this method preserves means and variances ($\sigma^2_{at_j} = \sigma^2_{at'_j}/(1+\alpha)$). The time complexity of noise addition is $\mathcal{O}(a \cdot n)$, being a the number of attributes which are protected.

Rank Swapping. Rank swapping [4] with parameter p and with respect to an attribute at_j (*i.e.* the j-th column of the original dataset X) can be defined as follows. Firstly, the records of X are sorted in increasing order of the values x_{ij} of the considered attribute at_j. For simplicity, we assume that the records are already sorted, that is $x_{ij} \leq x_{\ell j}$ for all $1 \leq i < \ell \leq n$. Then, each value x_{ij} is swapped with another value $x_{\ell j}$, randomly and uniformly chosen from the limited range $i < \ell \leq i + p$. When rank swapping is applied to a dataset, the algorithm explained above is run for each attribute to be protected, in a sequential way. If the number of attributes which are protected is a, the time complexity of rank swapping is $\mathcal{O}(a \cdot n \log n)$.

The parameter p is used to control the swap range. Normally, p is defined as a percent of the total number of records in X. Therefore, when p increases, the difference between x_{ij} and $x_{\ell j}$ may increase accordingly. This fact increases privacy, but of course the differences between the original and the protected dataset are higher, decreasing in this way statistical utility.

As noted in [18], the fact that each value is swapped with a value in a fixed, closed (and possibly public) rank makes this basic rank swapping method more prone to re-identification attacks, decreasing in this way the privacy that this method can offer. To mitigate this drawback, two variants of rank swapping are proposed in [18], where some values (with a small but still non-negligible probability) are swapped with values out of the theoretical rank. Despite these vulnerabilities against re-identification attacks, basic rank swapping is used by several European statistical agencies for data anonymization.

Resampling. Resampling [14] with parameter s and with respect to an attribute at_j can be described as follows. Firstly, the n records of X are sorted in increasing order of the values x_{ij} of the considered attribute at_j. Then, resampling takes s independent samples $X_1, \ldots, X_s$, each one containing n values of at_j, with replacement (i.e. with high probability, some values of at_j are taken more than once). Then, each sample is sorted in increasing order. Finally, the masked attribute is built by taking as first value the average of the first values of the samples, as second value the average of the second values, and so on. When resampling is applied to a dataset, the algorithm explained above is run for each attribute to be protected, in a sequential way. If a denotes the number of attributes which are protected, the time complexity of resampling is $\mathcal{O}(a \cdot s \cdot n \log n)$.

3 Rank Shuffling: A New SDC Method

As we have explained before, rank swapping has several interesting properties: it preserves univariate statistics, it is very handy and simple to explain and implement. However, it presents some drawbacks: the privacy level that it provides may decrease when the intruder uses the re-identification method presented in [18]. Furthermore, as we will show in Section 4.1, this method does not allow a secure multiparty version.

For these reasons, we propose in this section a new protection method called rank shuffling (rshuffling in short) that preserves the advantages of rank swapping, and disregards its drawbacks.

3.1 The Algorithm

As we have explained in Section 2.2, rank swapping swaps one original value with one of the p following values in the sorted table. Therefore, once the protected values of the attribute are published, as it happens in the SDC scenario, it is possible to restrict the protected records into which a specific original record may have been mapped [18]. Formally, the intruder must compare the original record x_i that he wants to link with only $2p$ records in the protected dataset (note that a protected value can be either the source or the destination in the swap process). In other words, for every original attribute value x_{ij}, there is an efficiently computable set $B(x_{ij})$ of $2p$ protected records which may be the result of transforming the original record x_i.

Obviously, if more than one attribute is known, it is possible to repeat the process for each attribute. In particular, if the original record x_i is represented by $x_i = (x_{i1}, \ldots, x_{ic})$ for c attributes $at_1, \ldots, at_c$, then the matching protected record x'_ℓ will necessarily satisfy the condition

$$x'_\ell \in \bigcap_{1 \leq j \leq c} B(x_{ij})$$

That is, the search of the linkage is reduced to the intersection of the sets of possible protected records. Of course, the more attributes are considered, the less records will be in this intersection, and therefore the probability of finding the correct record linkage will increase.

A possible solution for this problem is that any value in the dataset can be selected for a swap. However, the negative effect produced by swapping two very far values is that the information loss of the protected files increases. For this reason, in this paper we propose to replace the swapping step with a shuffling step, a random permutation. Formally, our new rank shuffling protection algorithm is defined in Algorithm 1. The first step of rank shuffling is to sort the records of X in increasing order considering the values of x_{ij} as the ranking criterion. Then, a shuffling window $[f, l]$ is defined. Initially, f is equal to 0 and l is equal to p, the window size. In rank shuffling, p can be defined as an integer value lower than n or as a percent of the total number of records in X. Next, all x_{ij} values with $f \leq i \leq l$ are randomly shuffled, and then the shuffling window is shifted s positions. Note that the parameter s, the window slide, has to be an integer number between 1 and p. This shuffling step is repeated while $l \leq n$. The whole procedure is run for each attribute to be protected, in a sequential way. The time complexity of rank shuffling is $\mathcal{O}\left(a \cdot \left(n \log n + p \cdot \frac{n-p}{s}\right)\right)$, being a the number of attributes which are protected.

3.2 Experimental Results

In order to compare the quality of our new rank shuffling method with the methods considered in this work (i.e. those described in Section 2.2), we have considered the Census

Algorithm 1. Rank Shuffling

```
Data: $X$: original dataset with $n$ records, $p$: window size, $s$: window slide
Result: $X'$: protected data set
begin
    foreach $at_j$ to be protected do
        Records of $X$ are sorted in increasing order of the values $x_{ij}$
        $f = 0, \quad l = p$
        While $l \leq n$
          Random_Shuffle($x_{fj}, \ldots, x_{lj}$)
          $f = f + s, \quad l = l + s$
    end
    return X
end
```

dataset, one of the two reference datasets proposed in the CASC project [2]. Census dataset was extracted from the U. S. Census Bureau [7] using the Data Extraction System (DES). This dataset contains 1080 records with 12 attributes. We have protected the Census dataset by using different parameterizations of noise addition ($\alpha = 0.1$ and $\alpha = 0.2$), rank swapping ($p = 5, 10, 15$), resampling ($s = 2, 4$) and rank shuffling ($p = 10, s = 8$ and $p = 25, s = 20$).

We have computed the score, a well-known comparison measure sketched in Section 2.2. Intuitively, the lower the score, the better the protection method is. A public and free implementation of the score is available at `http://ppdm.iiia.csic.es`. The results obtained using this web page are presented in Table 1. So, on the light of these results, our method performs slightly better than rank swapping, which is the protection method with the lowest scores in this table. The best score for rank shuffling is 20.26 whereas the best score obtained by rank swapping is 20.88. Furthermore, if the re-identification method presented in [18] was considered, the disclosure risk of rank swapping would increase, and so would its score. Rank shuffling is in some sense immune to this re-identification technique because the swapping step is replaced with a shuffling step, and therefore there is not a closed interval for the choice of the element that will be swapped with a given element. Summing up, we can say that rank shuffling is a realistic and very good alternative to the existing data protection methods.

Table 1. Score and execution time results

	IL	DR	$Score$	$Time$ (sec.)
noise0.1	18.47	46.50	32.49	0.013
noise0.2	38.11	25.16	31.64	0.014
rs.5	30.78	14.90	22.84	0.47
rs.10	36.71	5.92	21.31	0.47
rs.15	37.57	4.20	20.88	0.42
resampling.2	29.84	84.61	58.21	0.50
resampling.4	21.95	90.71	53.72	0.82
rsshuffle.10-8	36.32	7.45	21.89	0.29
rsshuffle.25-20	35.85	4.67	20.26	0.28

4 Distributed Privacy-Preserving SDC

Suppose now that we want to compute and release to the public a perturbed dataset $X' = \rho(X)$, but the original records of X are not stored in a single device. Instead, we have a set of entities $\mathcal{P} = \{P_1, \ldots, P_t\}$, and each P_i holds a part X_i of X, that is $X = X_1 \cup \ldots \cup X_t$. This partition of X can be: (i) *horizontal*, if each entity holds all the attributes of some record(s); (ii) *vertical*, if each entity holds some attribute(s) of all the records; (iii) *hybrid*, if each entity holds some attribute(s), maybe not always the same, of some record(s).

Of course, for all the SDC methods which work attribute-by-attribute (e.g. those in the rank swapping family, rank shuffling, resampling or noise addition), one can consider without loss of generality that databases have a single attribute, and therefore only horizontal partitions are meaningful. The entities agree to jointly obtain $X' = \rho(X)$, but they do not want to reveal anything about their private inputs X_i to the other entities. The ideal solution would have a trusted third party (TTP) receiving X_i from each P_i in a secret way (through a completely secure channel), generating the global X, applying $\rho(X) = X'$ and broadcasting X'. The goal is to design a solution which works without any TTP, but with the entities themselves exchanging some information and jointly computing X'. Such a real solution must offer as much privacy as the ideal solution with a TTP does. In other words, the information that an attacker obtains from an execution of this real protocol is the same that he can deduce only from his own secret input (if he is one of the entities) and from the result X' of the protocol.

We face therefore a *multiparty computation problem*. In such a problem, each entity holds a secret input x_i, and they want to jointly compute a value $f(x_1, \ldots, x_t)$, for a public function f, by keeping secret the values of the inputs. There are general results about multiparty computation which ensure that any function can be securely computed in such a distributed way. These generic solutions are, however, very inefficient, so the goal is to find more efficient solutions for some particular functions f. In this work, we address this problem when $f = \rho$ is a statistical disclosure control method.

4.1 First (Negative) Result: Rank Swapping Family

Before explaining the three distributed privacy-preserving methods for statistical disclosure control that we propose (for noise addition, rank shuffling and resampling), we give a negative result: a distributed privacy-preserving (or multiparty) version of any SDC method in the swapping family can never offer the desired level of privacy.

Indeed, let us consider for simplicity the case where X is horizontally partitioned (remember that swapping methods work attribute-by-attribute) between $t = 2$ entities, P_1 and P_2. Therefore, P_1 knows all the attributes of some original records, whereas P_2 knows all the attributes of the rest of records. Assume now that P_1 and P_2 jointly apply a distributed protocol which results in $X' = \rho(X)$, where ρ is a perturbation method which swaps pairs of values of the same attribute.

P_1 knows which records in X' correspond to his original records, because the confidential attributes have not been modified. Let i be the index of one of P_2's original records. If the protection method is secure, then P_1 should not have a high probability to obtain information on the confidential attributes of the record i, even after having obtained the original non-confidential attributes of this record. But, for each of these original non-confidential attributes x_{ij} of the record i, entity P_1 can look for it in X'. With reasonably high probability, x_{ij} is now placed in a record $i' \neq i$ of X' that corresponds to P_1. If this is the case, P_1 can look for his value $x_{i'j}$ in X', which for sure will be in the record i, because the applied method is a swapping one.

Once P_1 has found the protected record i where $x_{i'j}$ lies, he has re-identified the non-confidential attributes of some record belonging to P_2 with the corresponding confidential attributes, breaking in this way the privacy of the system. Of course, if the values of an attribute for different records have many repetitions, this method is less effective. On the other hand, running this attack for different attributes, the success probability for the attacker P_1 increases. The conclusion is that perturbation methods in the swapping family are not suitable for the scenario where the original database is distributed among several entities.

4.2 Basic Ingredients

In this section we explain some protocols which will be employed in the design of the distributed privacy-preserving versions of noise addition, rank shuffling and resampling. We assume that each entity P_i in the set $\mathcal{P} = \{P_1, \dots, P_t\}$ holds a share sk_i of a secret key sk for Paillier's cryptosystem, $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$, and that the matching public key is pk. The following subprotocols can all be proved secure, in the sense that they offer the same privacy level as the ideal solution with a TTP, assuming that Paillier's cryptosystem is secure. Similar subprotocols exist for the case where the initial private inputs are masked by using secret sharing schemes (see [5], for example).

Shuffled union of encrypted elements. Suppose that each entity P_i has as input a set of elements $A_i = \{a_{i,1}, \dots, a_{i,n_i}\}$. The output of this protocol is a set containing encryptions of all these elements $\{\mathcal{E}_{pk}(a_{i,j})\}_{1 \leq i \leq t, 1 \leq j \leq n_i}$, in a random and unknown order. The idea is to hide which elements correspond to each entity. We assume that the number n_i of elements of each entity is considered a private (sensitive) value. If this is not the case, easier protocols can be considered.

Let N be a public upper bound for n_i, and let h be a public function. Now each ciphertext have the form $(\mathcal{E}_{pk}(x_{i,j}), \mathcal{E}_{pk}(y_{i,j}), \mathcal{E}_{pk}(a_{i,j}))$. The protocol works as follows.

1. For the real elements $a_{i,j}$ with $j = 1, \dots, n_i$, each entity P_i chooses $x_{i,j}$ at random and computes $y_{i,j} = h(x_{i,j})$.
2. Each P_i completes his set A_i with $N - n_i$ "dummy" elements $a_{i,j}$, for $j = n_i + 1, \dots, N$. For these dummy elements, P_i chooses $x_{i,j}$ and $y_{i,j}$ at random, such that $y_{i,} \neq h(x_{i,j})$. Note that now all the sets A_i contain N elements.
3. P_1 encrypts all his values, $c_{1,j} = (\mathcal{E}_{pk}(x_{1,j}), \mathcal{E}_{pk}(y_{1,j}), \mathcal{E}_{pk}(a_{1,j}))$, for $j = 1, \dots, N$, and broadcasts the set C_1 which contains all these tuples of ciphertexts.
4. For $i = 2, \dots, t$, entity P_i re-randomizes the ciphertexts in C_{i-1}, then adds his own encryptions $c_{i,j} = (\mathcal{E}_{pk}(x_{i,j}), \mathcal{E}_{pk}(y_{i,j}), \mathcal{E}_{pk}(a_{i,j}))$, for $j = 1, \dots, N$, and finally applies a random permutation to the resulting set of tuples of ciphertexts. The set resulting from this permutation is defined as C_i, and is broadcast.
5. P_1 applies a last random permutation to the tuples in C_t, and broadcasts the resulting set C of tuples of ciphertexts.

6. For each tuple in C, all the entities together run the decryption algorithm $\mathcal{D}$ to the first two ciphertexts of each tuple, obtaining $x_{i,j}$ and $y_{i,j}$. The final output of the protocol, C, contains the third ciphertexts of the tuples for which $y_{i,j} = h(x_{i,j})$.

We will denote an execution of this protocol as $C \leftarrow \texttt{Union}(\{a_{i,j}\}_{1 \le i \le t, 1 \le j \le n_i})$.

Multiplying two encrypted values. Remember that Paillier's cryptosystem is additively homomorphic, so everyone can obtain an encryption $\mathcal{E}_{pk}(a + b)$ given encryptions $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$. We want to do the same with multiplication: the input will be $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$, and the desired output is $\mathcal{E}_{pk}(ab)$. The following protocol for this primitive was proposed in [3].

1. Each entity P_i chooses at random a value d_i and broadcasts $\mathcal{E}_{pk}(d_i)$. We denote $d = \sum_{1 \le i \le t} d_i$.
2. Using the homomorphic properties, they can compute $\mathcal{E}_{pk}(a + d)$. They jointly run $\mathcal{D}$ to decrypt this value, obtaining $a + d$.
3. P_1 chooses $a_1 = a + d - d_1$. For $i = 2, \ldots, t$, each P_i sets $a_i = -d_i$. Note that $a = \sum_{1 \le i \le t} a_i$.
4. Each P_i broadcasts $\mathcal{E}_{pk}(a_i b)$, which is computed by $\oplus$-operating $\mathcal{E}_{pk}(b)$ with itself a_i times.
5. Finally, one of the entities can compute

$$\bigoplus_{1 \le i \le t} \mathcal{E}_{pk}(a_i b) = \mathcal{E}_{pk}(b \sum_{1 \le i \le t} a_i) = \mathcal{E}_{pk}(ba).$$

We will denote an execution of this protocol as $\mathcal{E}_{pk}(ab) \leftarrow \texttt{Multip}(\mathcal{E}_{pk}(a), \mathcal{E}_{pk}(b))$.

Obtaining encryptions of the bits of an encrypted element. The input of this protocol is a Paillier encryption $\mathcal{E}_{pk}(a)$ of some element a. If $(a_{\ell-1}, \ldots, a_1, a_0) \in (\mathbb{Z}_2)^\ell$ is the bit decomposition of a (i.e. $a = \sum\limits_{0 \le i \le \ell-1} a_i 2^i$), then the desired output is a tuple $(\mathcal{E}_{pk}(a_{\ell-1}), \ldots, \mathcal{E}_{pk}(a_1), \mathcal{E}_{pk}(a_0))$ containing Paillier encryptions of the bits in the representation of a.

Schoenmakers and Tuyls [20] have presented solutions to this problem. They present a general solution for the case where a can be any element in the plaintext space of Paillier's cryptosystem (i.e. a is 1024 bits long), and then a more efficient solution for the case where a is much smaller (which will be the case in our protocols, since a will be a value of the database).

We will denote an execution of this protocol as $(\mathcal{E}_{pk}(a_{\ell-1}), \ldots, \mathcal{E}_{pk}(a_1), \mathcal{E}_{pk}(a_0)) \leftarrow \texttt{Bits}(\mathcal{E}_{pk}(a))$.

Comparing two encrypted values. Given two encryptions $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$ as input, we want to have as output an encryption of a bit indicating which value is greater; namely, the output will be $\mathcal{E}_{pk}(1)$ if $a < b$, and will be $\mathcal{E}_{pk}(0)$ if $a \ge b$. To implement this primitive, we use and simplify some of the techniques that appear in [5]. The idea is to consider the bit representations of a and b, and then to find the highest position j in which $a_j \ne b_j$ (or, in other words, where a_i XOR $b_i = 1$). The output $\mathcal{E}_{pk}(b_j)$ is the desired one. Since the only information that we want to be leaked is if $a < b$ or not, the position j must be kept secret, as well, which makes the protocol more complicated. We assume that a and b have the same bit length ℓ. The protocol works as follows.

1. Run $(\mathcal{E}_{pk}(a_{\ell-1}), \ldots, \mathcal{E}_{pk}(a_1), \mathcal{E}_{pk}(a_0)) \leftarrow \texttt{Bits}(\mathcal{E}_{pk}(a))$ and $(\mathcal{E}_{pk}(b_{\ell-1}), \ldots, \mathcal{E}_{pk}(b_1), \mathcal{E}_{pk}(b_0)) \leftarrow \texttt{Bits}(\mathcal{E}_{pk}(b))$.
2. For each $i = 0, \ldots, \ell-1$, compute $\mathcal{E}_{pk}(e_i) = \mathcal{E}_{pk}(a_i \text{ XOR } b_i)$, by first computing $\mathcal{E}_{pk}(a_i - b_i)$ and then running $\mathcal{E}_{pk}(a_i \text{ XOR } b_i) \leftarrow \texttt{Multip}(\mathcal{E}_{pk}(a_i - b_i), \mathcal{E}_{pk}(a_i - b_i))$.
3. Compute the (encrypted version of the) vector of OR-prefixes f_i of $e_i{=}a_i$ XOR b_i, that is, $f_i = \bigvee_{j=i}^{\ell-1} e_i$. Note that this is equivalent to $f_{\ell-1} = e_{\ell-1}$ and $f_i = f_{i+1} \vee e_i$, for $i = \ell-2, \ldots, 1, 0$. Taking into account that $f_{i+1} \vee e_i = f_{i+1} + e_i - f_{i+1}e_i$, we can compute $\mathcal{E}_{pk}(f_i)$ as $\mathcal{E}_{pk}(f_{\ell-1}) = \mathcal{E}_{pk}(e_{\ell-1})$ and, for $i = \ell-2, \ldots, 1, 0$:
$$\mathcal{E}_{pk}(f_i) = \mathcal{E}_{pk}(f_{i+1}) + \mathcal{E}_{pk}(e_i) - \texttt{Multip}(\mathcal{E}_{pk}(f_{i+1}), \mathcal{E}_{pk}(e_i)).$$
4. Define $\mathcal{E}_{pk}(g_{\ell-1}) = \mathcal{E}_{pk}(f_{\ell-1})$. For $i = \ell-2, \ldots, 1, 0$, compute $\mathcal{E}_{pk}(g_i) = \mathcal{E}_{pk}(f_i) - \mathcal{E}_{pk}(f_{i+1})$.
5. For $i = \ell-1, \ldots, 1, 0$, run $\mathcal{E}_{pk}(h_i) \leftarrow \texttt{Multip}(\mathcal{E}_{pk}(g_i), \mathcal{E}_{pk}(b_i))$.
6. Output $\mathcal{E}_{pk}(z) = \sum_{i=0}^{\ell-1} \mathcal{E}_{pk}(h_i)$.

Note that, if j is the highest position where $a_j \neq b_j$, then we have $f_i = 1$ if and only if $i \leq j$. Therefore, we have $g_j = 1$ and $g_i = 0$ for $i \neq j$, as desired. We will denote an execution of this protocol as $\mathcal{E}_{pk}(z) \leftarrow \texttt{Compare}(\mathcal{E}_{pk}(a), \mathcal{E}_{pk}(b))$.

4.3 Distributed Privacy-Preserving Noise Addition

The main difficulty that one finds when designing a distributed version of the random noise addition method is the computation of the variance of the attributes. Given a list of n values $\{x_{ij}\}_{1 \leq i \leq n}$ of some attribute at_j, the variance of this attribute is

$$\sigma^2_{at_j} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \overline{x})^2 = \frac{1}{(n-1)n^2} \sum_{i=1}^{n} (nx_{ij} - n\overline{x})^2 \ ,$$

where $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$ is the mean of the n values.

Remember that the variance of a perturbed attribute is $1 + \alpha$ times the variance of the original attribute. Assuming that the parameter α is public, then the variance of the original attributes will be known once the perturbed dataset is released, so broadcasting these variances during the execution of the multiparty protocol is not a privacy breach. Something similar happens with the mean $\overline{x}$ and the total number of records n.

Let A_ℓ denote the set of n_ℓ indices of the records that belong to entity $P_\ell \in \{P_1, \ldots, P_t\}$. We use a threshold homomorphic encryption scheme $\mathcal{E}$ (such as Paillier's) with public key pk. To jointly compute the variance of an attribute at_j, the entities run the following protocol.

1. Each entity P_ℓ computes and broadcasts the ciphertexts
$$c_{1,\ell} = \mathcal{E}_{pk}\left(\sum_{i \in A_\ell} x_{ij}\right), \qquad c_{2,\ell} = \mathcal{E}_{pk}(n_\ell).$$

2. One entity computes $C_1 = \bigoplus_{1\le\ell\le t} c_{1,\ell}$ and $C_2 = \bigoplus_{1\le\ell\le t} c_{2,\ell}$.
3. They jointly decrypt C_1 and C_2, obtaining $\sum_i x_{ij}$ and n. They divide the results, to obtain $\overline{x}$.
4. Each entity P_ℓ computes and broadcasts the ciphertext

$$\tilde{c}_\ell = \mathcal{E}_{pk}\left(\sum_{i\in A_\ell} (nx_{ij} - n\overline{x})^2\right).$$

5. One entity computes $\tilde{C} = \bigoplus_{1\le\ell\le t} \tilde{c}_\ell$.
6. They jointly decrypt $\tilde{C}$, and then they divide the result by $(n-1)n^2$. They output the resulting value as $\sigma^2_{at_j}$.

We want to stress that the plaintext space of Paillier's cryptosystem is $\mathbb{Z}_{\tilde{N}}$, where $\tilde{N}$ is a product of two secret prime numbers, and $\tilde{N}$ is at least 1024 bits long. Therefore, assuming that the values in the database are bounded by $2^{512}/n\sqrt{n}$ (which is a very weak assumption), we have that all the decrypted values in steps 3 and 6 are the desired values (i.e. there is no modular reduction).

Now each entity can individually perturb his values, by computing $x'_{ij} = x_{ij} + \epsilon$, where ϵ is a normally distributed error drawn from a random variable $\epsilon \sim N(0, \alpha\sigma^2_{at_j})$.

In order to broadcast the resulting perturbed data in a private way, without leaking which records belong to each entity, the entities jointly execute $C \leftarrow \texttt{Union}(\{x'_i\}_{1\le\ell\le t, i\in A_\ell})$ and jointly decrypt the resulting ciphertexts, to obtaining the perturbed dataset X'.

4.4 Distributed Privacy-Preserving Rank Shuffling

To design a secure multiparty version of our new perturbative method, rank shuffling, we need to use all the basic ingredients described in Section 4.2. Since rank shuffling works attribute-by-attribute, let us assume that the original database X is horizontally partitioned among t entities $P_1, \ldots, P_t$. Let A_ℓ denote the set of indices of the records that belong to entity P_ℓ. Let pk be the public key of the employed threshold homomorphic encryption scheme $\mathcal{E}$. Let p, s be the public parameters for rank shuffling: p is the window size, and s is the window slide. The distributed protocol is as follows.

1. Each P_ℓ computes, for each record $i \in A_\ell$, the tuple $(\{\mathcal{E}_{pk}(x_{ij})\}_{1\le j\le V})$ containing the encryptions of all the attributes (both confidential and non-confidential) of record i. We denote these vectors of encryptions as $\boldsymbol{c}_i = (c_{i1}, \ldots, c_{iV})$.
2. Run $C \leftarrow \texttt{Union}(\{\boldsymbol{x}_i\}_{1\le\ell\le t, i\in A_\ell})$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iV})$.
3. For each (non-confidential) attribute at_j to be protected, do the following.

   (a) Making calls to the `Compare` algorithm, sort the table C in increasing order with respect to the attribute at_j.
   (b) Define $f = 0$ and $l = p$.

(c) While $l \leq n$ do:

- Re-randomize and permute the values $\{c_{fj}, \ldots, c_{lj}\}$.
- $f = f + s, \qquad l = l + s$.

4. Each P_ℓ re-randomizes and permutes the resulting vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n$.
5. Decrypt jointly all the ciphertexts in the resulting table C.

The most costly part of this protocol is step 3(a), which involves $n \log n$ calls to the `Compare` algorithm, which is itself quite expensive. Note that we also encrypt the confidential attributes, even if they are not going to be perturbed. If this was not done, then the sorting which is performed in step 3(a) would reveal potentially private information: the relation between the confidential attributes and the low (and medium, and high) values of the corresponding non-confidential attribute at_j.

Step 4 has the goal of hiding the last sorting. If this step was not executed, then step 5 (global decryption) would reveal the confidential attributes which correspond to the records with lower (and medium, and higher) values of the last perturbed attribute.

4.5 Distributed Privacy-Preserving Resampling

Again, we assume that the original database X is horizontally partitioned among t entities $P_1, \ldots, P_t$. Let A_ℓ denote the set of indices of the records that are known to entity P_ℓ. Remember that we will use a a threshold homomorphic encryption scheme $\mathcal{E}$ (such as Paillier's) with public key pk. The algorithm works as follows.

1. Each P_ℓ computes, for each record $i \in A_\ell$, the tuple $(\{\mathcal{E}_{pk}(x_{ij})\}_{1 \leq j \leq V})$ containing the encryptions of all the attributes (confidential and non-confidential) of record i.
2. Run $C \leftarrow \texttt{Union}(\{\boldsymbol{x}_i\}_{1 \leq \ell \leq t, i \in A_\ell})$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iV})$.
3. For each (non-confidential) attribute at_j to be protected, do the following.

   (a) Making calls to the `Compare` algorithm, sort the table C in increasing order with respect to the attribute j.
   (b) Take s independent samples, with replacement, of the attribute j of table C, each one with size n. We denote these samples as $C_1^{(j)}, \ldots, C_s^{(j)}$.
   (c) Making calls to the `Compare` algorithm, sort each of the samples, increasingly.
   (d) For $i = 1, \ldots, n$, replace the i-th value of the attribute at_j in the sorted version of C with the sum of the i-th values of the sorted versions of $C_1^{(j)}, \ldots, C_s^{(j)}$.

4. Each P_ℓ re-randomizes and permutes the resulting table C.
5. Decrypt jointly all the ciphertexts in the resulting table C.
6. For those attributes at_j where resampling has been applied, divide the values in these columns with s.

Steps 3(b), 3(d) and 6 can be done by one of the entities alone. Note that the average step of resampling is now executed in two phases, step 3(d) for the sum and step 6 for the division by s. Remember that the samples $C_1^{(j)}, \ldots, C_s^{(j)}$ contain Paillier's ciphertexts. Everybody can therefore compute an encryption of the sum of the encrypted values, by using the homomorphic properties of Paillier's scheme. Since Paillier's plaintext space is 1024 bits long, and the values in the dataset are assumed to be much smaller, these sums will not be reduced modulo the plaintext space size, when working on encrypted data. Therefore, the values obtained in step 5 are exactly the sums of the considered sampled elements. In the last step, the average computation can be completed by dividing these sums by s.

4.6 Brief Discussion on Privacy

The fact that any distributed implementation of a method in the rank swapping family is not secure is inherent to the statistical protection method itself, not to the multiparty techniques. Actually, even an ideal implementation of such a distributed method, with the participation of a TTP, would be insecure: the same attacks as those explained in Section 4.1 would apply.

For the other methods considered in this work (noise addition, resampling, rank shuffling), this problem does not exist: an ideal solution implemented by a TTP would achieve a satisfactory level of privacy. Then, the multiparty implementations that we propose here do not decrease this level of privacy, because all the subprotocols that are used (ordering, comparing, extracting bits) do not leak any private information, as long as the employed threshold homomorphic encryption scheme is secure.

5 Conclusions and Future Work

We have presented in this work the first detailed study about distributed privacy-preserving methods for statistical data perturbation. We have first shown that secure multiparty versions of some perturbation methods (in the swapping family) are not possible. We have then proposed a new data perturbation method, rank shuffling, which obtains similar quality results (score) than rank swapping, while being more robust against re-identification attacks. Finally, we have designed distributed privacy-preserving protocols for noise addition, rank shuffling and resampling.

As future work, we can first mention microaggregation [6]. It seems quite trivial to extend existing multiparty protocols [15,1] in order to design secure distributed versions of some microaggregation algorithms. For more complicated microaggregation algorithms, where the size of the (intermediate) clusters can vary and is therefore a sensitive value, the only existing satisfactory solution for the multiparty case [1] works only when the number of entities is $t = 2$. Finally, regarding the new method of rank shuffling, it would be interesting to analyze it in a more formal and detailed way, from both a theoretical point of view (using order statistics, or discussing what level of differential privacy [11] it achieves) and a practical point of view (running it on other databases, of different kinds and sizes, comparing execution times, etc.).

Acknowledgements

Partial support by the Spanish MEC (projects ARES – CONSOLIDER INGENIO 2010 CSD2007-00004 – and eAEGIS – TSI2007-65406-C03-02, and grant *Ramón y Cajal* for J. Herranz) and by the European Community (Marie Curie fellowship No 235226 for J. Nin) is acknowledged.

References

1. Bunn, P., Ostrovsky, R.: Secure two-party k-means clustering. In: Proc. of CCS 2007, pp. 486–497. ACM Press, New York (2007)
2. CASC: Computational Aspects of Statistical Confidentiality, European Project IST-2000-25069, http://neon.vb.cbs.nl/casc
3. Cramer, R., Damgård, I.B., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 280–299. Springer, Heidelberg (2001)
4. Dalenius, T., Reiss, S.P.: Data-swapping: a technique for disclosure control. Journal of Statistical Planning and Inference 6, 73–85 (1982)
5. Damgård, I.B., Fitzi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 285–304. Springer, Heidelberg (2006)
6. Defays, D., Anwar, M.N.: Micro-aggregation: a generic method. In: Proc. of the 2nd International Seminar on Statistical Confidentiality, pp. 69–78 (1995)
7. Data Extraction System, U.S. Census Bureau, http://www.census.gov
8. Domingo-Ferrer, J., Torra, V.: Disclosure control methods and information loss for microdata. In: [10], pp. 91–110 (2001)
9. Domingo-Ferrer, J., Torra, V.: A quantitative comparison of disclosure control methods for microdata. In: [10], pp. 111–133 (2001)
10. Doyle, P., Lane, J., Theeuwes, J., Zayatz, L. (eds.): Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies. Elsevier Science, Amsterdam (2001)
11. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
12. Dwork, C., Yekhanin, S.: New efficient attacks on statistical disclosure control mechanisms. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 469–480. Springer, Heidelberg (2008)
13. Fouque, P.A., Poupard, G., Stern, J.: Sharing decryption in the context of voting or lotteries. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 90–104. Springer, Heidelberg (2001)
14. Heer, G.R.: A bootstrap procedure to preserve statistical confidentiality in contingency tables. In: Proc. of the 1st International Seminar on Statistical Confidentiality, pp. 261–71 (1993)
15. Jagannathan, G., Wright, R.: Privacy-preserving distributed k- means clustering over arbitrarily partitioned data. In: Proc. of KDD 2005, pp. 593–599 (2005)
16. Kim, J.J.: A method for limiting disclosure in microdata based on random noise and transformation. In: Proc. of the ASA Section on Survey Research Methodology, pp. 303–308 (1986)
17. Lane, J., Heus, P., Mulcahy, T.: Data access in a cyber world: making use of cyberinfrastructure. Transactions on Data Privacy 1(1), 2–16 (2008)
18. Nin, J., Herranz, J., Torra, V.: Rethinking rank swapping to decrease disclosure risk. Data & Knowledge Engineering 64(1), 346–364 (2008)
19. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
20. Schoenmakers, B., Tuyls, P.: Efficient binary conversion for Paillier encrypted values. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 522–537. Springer, Heidelberg (2006)

Towards a Privacy-Preserving National Identity Card

Yves Deswarte[1,2] and Sébastien Gambs[1,2,⋆]

[1] CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France
[2] Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
{Yves.Deswarte,Sebastien.Gambs}@laas.fr

Abstract. In this paper, we propose to replace the national identity card, currently used in many countries, by a personal device that allows its user to prove some binary statements about himself while minimizing personal information leakage. The privacy of the user is protected through the use of anonymous credentials which allows him to prove binary statements about himself to another entity without having to disclose his identity or any unnecessary information. The proposed scheme also prevents the possibility of tracing the user, even if he proves several times the same statement (unlinkability property). A tamper-proof smartcard is used to store the personal information of the user thus protecting his privacy and preventing the risks of forgery at the same time. The user identifies himself to the card via biometrics thus forbidding an unauthorized use in the situation where the card is stolen or lost. Two practical implementations of the privacy-preserving identity card are described and discussed.

1 Introduction

Intuitively, respecting the principles of *data minimization*[1] and *data sovereignty*[2] when using a national identity card seems to be at odds with other obligations required in practical tasks from everyday life such as checking the nationality of the owner of the card when he crosses a border, verifying his age when he wants to obtain some discount related to it or proving that he belongs (or does not

[⋆] This research was performed when Sébastien Gambs was CNRS postdoctoral researcher at LAAS. Since September 2009, he has moved to IRISA (Rennes) on a joint research chair between Université de Rennes 1 and INRIA (sgambs@irisa.fr).

[1] The data minimization principle states that only the information necessary to complete a particular application should be disclose (and no more). This principle is a direct application of the legitimacy criteria defined by the European data protection directive (Article 7, [14]).

[2] The data sovereignty principle states that the data related to an individual belong to him and that he should stay in control of how these data are used and for which purpose. This principle can be seen as an extension of many national legislations on medical data that consider that a patient record belongs to the patient, and not to the doctor that creates or updates it, nor to the hospital that stores it.

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 48–64, 2010.

belong) to a particular group. In this paper, we advocate that this intuition is wrong by introducing the concept of *privacy-preserving identity card*.

Definition 1 (Privacy-preserving Identity Card). *A* privacy-preserving id-entity card *is a personal device that allows its user*[3] *to prove some binary statements about himself (such as his right of access to some resources) while minimizing personal information leakage.*

Our proposal for the privacy-preserving national identity card is close in spirit to the project PRIME[4] (PRivacy and Identity Management for Europe) [20], whose goal was to develop a framework and tools allowing a user to manage his identity and to protect his privacy in the cyberspace. Indeed, the main purpose of the privacy-preserving identity card is to enable a person to conduct tasks in the real world without having to disclose his identity whereas PRIME was focusing exclusively on the online setting. Birch's informal proposition, called Psychic ID, for the future U.K. national identity card [3], also shares several privacy features with our proposal. Indeed, the Psychic ID card respects the principle of data minimization and only reveals to a reader (or visually to an entitled person) the minimal information concerning the user that is needed for a specific purpose if the user possesses the corresponding credential, and nothing otherwise. An overview of the privacy features of the specifications of the future European electronic identity cards can be found in [15]. Other related works close to our approach include a protocol for the partial revelation of information related to certified identity proposed by Boudot [4] and the development of a cryptographic framework for the controlled release of certified data due to Bangerter, Camenisch and Lysyanskaya [1].

The outline of the paper is the following. First in Section 2, we detail in an abstract way the desirable properties that a privacy-preserving identity card should fulfill. Afterwards in Section 3, we briefly review some enabling technologies on smartcards, anonymous credentials and biometric authentication that will be the basis of our practical implementations of the card. Then in Section 4, we briefly describe how such a card might be use in practice before in Section 5, proposing two practical implementations of the privacy-preserving identity card. Finally, we conclude in Section 6 with a discussion on possible extensions to the privacy-preserving identity card. An extended version of this paper is available at [12].

2 Desiderata for a Privacy-Preserving Identity Card

In this paper, we adopt a notation inspired from the work of Camenisch and Lysyanskaya on *anonymous credentials* [6] (see Section 3.2 for more details). In particular, we call the owner of the privacy-preserving identity card, the *user*

[3] In this paper, we use the word "user" to denote at the same time both the owner and the effective user of the card. Indeed as the user needs to authenticate to the card before he can use it, the user of the card will effectively always be also his owner.

[4] https://www.prime-project.eu/

(who is likely to be a simple citizen). The *Registration Authority* (RA) is a legal entity (such as the city hall) that can check the personal information of the user and register the request for a privacy-preserving identity card. The *Certification Authority* (CA) is a trusted third party (for instance the government) that will sign the information transmitted by the RA to certify its validity. Once the card has been issued, the RA and CA are no longer involved in the picture except if the user needs a new card or if there is a valid reason for lifting the anonymity of the user. An *organization* is an entity that can grant access to some of its resources to the user. (For example, an organization could be the immigration services, a theater or an airline company.) A *verifier* belongs to one organization and interacts with the user to check his right of access to the resources of this organization. In practice, the verifier is usually a smartcard reader device connected to the network of the organization that can communicate with the privacy-preserving identity card. Ideally, the privacy-preserving identity card should fulfill at least the following properties:

- *No personal information leakage*: in order to protect the privacy of the user, the card should disclose as little information as possible about him. Ideally, the only thing the card should reveal is one bit of information proving (or disproving) a binary statement concerning the user.
- *Unlinkability*: it should not possible to trace and link the actions of the user of the card. For instance, even if the user proves the same statement at different occasions, it should be impossible to link the different statements as being made by the same user.
- *Ownership proof*: only the legitimate user should be able to use his privacy-preserving identity card to prove statements about himself to other entities. This means that some authentication mechanism has to take place between the user and the card. The purpose of this authentication step is to avoid an unauthorized use of the card. This authentication mechanism should also guarantee the *non-transferability* of the card. Otherwise, the user could sell for some money the use of his privacy-preserving identity card to somebody else, thus transferring his privileges or even his identity to illegitimate users.
- *Authenticity*: some mutual authentication has to be performed between the card and the reader device in order to prevent the possibility of an adversary impersonating the role of a valid privacy-preserving identity card or a valid reader. This authentication will assert the authenticity of both the card and the reader.
- *Correctness*: a binary statement proven by the user with the help of the privacy-preserving identity card should always be valid. For instance, the user should never be able to prove false statements about himself by cheating the system (*soundness property*). Moreover if the verifier is honest, he should always accept a binary statement about the user provided that this statement is true and the user possesses the corresponding credentials (*completeness property*).
- *Unforgeability*: in order to avoid someone counterfeiting the identity card and usurping the role of the user, the card should be tamper-proof and have an inherent ability to resist hardware and logical attacks.

Apart from these fundamental requirements, the privacy-preserving identity card may also respect some additional properties such as:

- *Optional anonymity removing*: the actions of the user should stay anonymous at all times, except in some scenarios where it might be necessary to remove his anonymity for serious reasons. For instance in an extreme situation, it could happen that a crime (such as a murder) has been perpetrated in a room that has been accessed by only one person using a privacy-preserving identity card. In this situation, the certification authority and the verifier may want to collaborate in order to lift the anonymity of this person. On the other hand, although the possibility of lifting the anonymity is desirable in some scenarios, it could decrease the confidence of the user in his belief that his privacy will really be protected by the card.
- *Explicit consent*: in order to increase the trust of the user in the system, the card could monitor the questions that it has been asked and display them to the user. It is even possible to imagine, that for some questions that are deemed critical regarding the privacy of the user, his confirmation may be asked before the privacy-preserving identity card replies the question.

3 Enabling Technologies

Enforcing in reality the properties of the privacy-preserving identity card requires the combination of several hardware and cryptographic techniques that we briefly review in this section.

3.1 Smartcards

A *smartcard* is a plastic card with an embedded integrated circuit that contains some dedicated memory cells and a microprocessor that can process data stored in the memory cells or exchanged with a reader through serial link connections (for contact smartcards), or through radio links (for contactless smartcards). The memory cells can only be accessed by the microprocessor. The main purpose of the smartcard is to assure the confidentiality and integrity of the information stored on the card. For that, the smartcard must satisfy inherent tamper-proof properties (to protect the microprocessor and the memory) as well as some resistance against physical attacks and side-channel analysis[5]. As in cryptology, there is a ongoing race in the smartcard world between the developers of attacks and the designers of counter-measures (see [21] for instance).

Nowadays, smartcards are widely used around the world, especially in mobile phones, tokens for public transport systems or for other applications such as electronic payments. Until now, smartcards used in practice have relied mostly on symmetric encryption (by using algorithms such as triple DES or CRYPTO-1)[6].

[5] The same kind of tamper-proofness techniques can be applied to USB keys, smartcard readers or other hardware devices for similar purposes.

[6] This assertion is true at least for low-cost smartcards, even if public-key cryptosystems are available on many recent smartcards, including JavaCards.

Calmels, Canard, Girault and Sibert have recently suggested to move instead to asymmetric encryption in the future for RFID tags, both for security and practical reasons [5]. They have also described a low-cost version of a group signature scheme thus demonstrating that such cryptographic primitive are within the reach of inexpensive smartcard technologies.

3.2 Anonymous Credentials

An *anonymous credential* is a cryptographic token which allows a user to prove statements about himself anonymously to verifiers. Anonymous credentials are generally based on *zero-knowledge* proofs [16] and enable the user to prove his accreditation to the verifier without revealing any additional information (such as his identity). The first system of anonymous credential is due to Chaum [9] and is based on the idea that each organization might know the same user by a different *pseudonym*. The organizations cannot combine their data on a particular user because they are unable to link two different pseudonyms to the same person. *Private credentials* can be derived from other credentials and used to prove relationships between credentials/attributes/organizations without having the risk of linking the different pseudonyms.

Credentials can be *one-show* (as it is the case for e-cash) or *multiple shows*. When a user shows multiple times the same credential, this raises the concern of linkability if several actions can be traced to a unique user (even anonymous). One possibility for preventing this is to issue multiple one-show credentials to the same user. Another solution is to use a *group signature scheme* which allows multiple-show unlinkability. Group signature schemes [10] have been introduced by Chaum and van Heyst to provide anonymity to the signer of the message. For that, there is a single public verification key for the group, but each member of the group receives a different private signing key from the group manager (who could be for instance the CA). A group signature scheme (with optional anonymity removing) consists in general of the four following operations:

- *Registration of the user*. During the Join operation, the CA assigns to the user a new private signature key, which we denote by SKG_U.
- *Signature of a message in behalf of the group*. The $\mathsf{SignGroup}$ operation takes as input a message m and signing key SKG_U and produces a signature $\sigma_{G,U}(m)$ on this message.
- *Verification of a group signature*. The $\mathsf{VerifySignGroup}$ operation allows to check the validity of a group signature. It requires as input a verification key for the group VKG, which has been setup by the CA and is publicly known, as well as a message m and a group signature on this message $\sigma_{G,U}(m)$. $\mathsf{VerifySignGroup}$ produces as output either accept or reject depending on the validity of the signature.
- *Anonymity removing*. From the point of view of the verifier, it is impossible to distinguish if two group signatures come from the same individual or not. However in exceptional situations, the CA can (in association with the verifier) retrieve the identity of a particular signer via the $\mathsf{LiftAnonymity}$

operation. This operation takes as input a message m and a group signature on this message $\sigma_{G,U}(m)$ and produce as output the identity of the signer U. In practice, this is often done by first finding the corresponding signature key SKG_U and then retrieving the identity associated to this key.

Another possibility for implementing anonymous credentials is to use a *non-interactive zero-knowledge proof* [2] in combination with a *commitment scheme*. A commitment scheme is characterized by two operations:

- *Commitment phase.* During this phase, the Commit operation takes as input a value a and some auxiliary information aux (which corresponds generally to some form of randomness) and produces $comm(a)$ which is a commitment to this particular value a.
- *Opening phase.* The Open operation takes as input a commitment $comm(a)$ and the information aux and reveals as output a, the committed value.

A commitment scheme is *perfectly binding* if there is only one a that corresponds to a particular commitment $comm(a)$ (i.e., an adversary cannot open a commitment to several values), and *computationally hiding* if an adversary with bounded computational power cannot open a particular commitment without access to the auxiliary information. Suppose that a prover stores a particular value a and the CA's signature on it, $\sigma_{CA}(a)$, which certifies its validity. The prover may want to show that this value respects a particular binary statement f to a verifier in a zero-knowledge manner. To realize that, the prover sends to the verifier $comm(a) \leftarrow \mathsf{Commit}(a, aux)$, which is a commitment to the value a. Then, the prover issues $\pi \leftarrow \mathsf{Prove}((a, \sigma_{CA}(a), aux)|\mathsf{VerifySign}(a, \sigma_{CA}(a), VK_{CA}) = \mathsf{accept} \wedge a = \mathsf{Open}(comm(a), aux) \wedge f(a) = true)$, which is a non-interactive zero-knowledge proof that the prover knows (a, $\sigma_{CA}(a)$,aux) such that (1) $\sigma_{CA}(a)$ is a valid signature of the CA on a (verified by VK_{CA}, the public verification key of the CA); and (2) the committed value of $comm(a)$ is effectively a; and (3) the value a respects the binary statement f.

3.3 Biometric Authentication

The *biometric profile* of a person is composed of a combination of some physical features that uniquely characterize him. For instance, a biometric feature can be a fingerprint or a picture of the iris. The biometric data of an individual is a part of his identity just as his name or his address. Such biometrics can be used for the purpose of *identification* (i.e., identifying a particular individual in a list of registered people) or *authentication* (verifying that the person claiming an identity is indeed the one who has been registered with this identity).

In order to verify that an individual corresponds to some registered biometric profile, a fresh sample of his biometric data is generally taken and compared with the stored template using a matching algorithm. The matching algorithm computes a dissimilarity (or distance) measure[7] that indicates how far are the

[7] The dissimilarity measure used can be for instance the Hamming distance, the set difference or the edit distance.

two biometric samples. Two biometric samples are considered to belong to the same individual if their dissimilarity is below some well-chosen threshold, which is dependent of the natural variability within the population. A good biometric strategy tries to find a compromise between false acceptance rate or FAR (wrongly recognizing the individual as a particular registered user) and the false rejection rate or FRR (being unable to recognize the registered user). An example of biometric data is the picture of the iris that can be transformed/coded into a vector of 512 bytes called the IrisCode. Afterwards, it is fairly simple to evaluate the dissimilarity between two codewords simply by computing the Hamming distance between these two vectors. In practice, this method can lead to very low rates of false acceptance ($< 0.1\%$) and false rejection ($< 1\%$).

As the biometric features of an individual is an inherent part of his identity, several techniques have been developed to avoid storing explicitly the biometric profile while keeping the possibility of using it for authentication. For instance, some techniques have been proposed which combine the use of error-correcting codes and hash function such as the *fuzzy commitment scheme* [18]. In the same spirit as the fuzzy commitment scheme, a cryptographic primitive known as *fuzzy extractor* has been developed in the recent years (see for instance the survey [13]). Let b be the biometric profile of the user[8]. This primitive allows to extract a uniformly distributed random string $rand \in \{0,1\}^l$ [9] from a biometric template b in a noise-tolerant manner such that if the input changes to some b' close to b (i.e. $dist(b, b') < t$), the string $rand$ can still be recovered exactly. When initialized for the first time, a fuzzy extractor outputs a helper string called $p \in \{0,1\}^*$, which will be part of the input of subsequent calls to the fuzzy extractor in order to help in reconstructing $rand$. The string p has the property that it can be made public without decreasing the security of $rand$. Formally, a fuzzy extractor consists of two operations:

- *Generation phase.* During the first use of the fuzzy extractor, the operation Generate takes as input a biometric template b and produces as output a uniform random string $rand$ and a helper string p.
- *Retrieval phase.* The operation Retrieve takes as input a biometric profile b' which is close to the original profile b (i.e. $dist(b, b') < t$) as well as the helper string p and produces as output the random string $rand$.

Fuzzy extractors can be used for biometric verification in a straightforward way. First, $h(rand)$ and p are stored inside the card during its creation after the Generate operation (for h a randomly chosen hash function). Afterwards, when the card wants to ensure that the current user is indeed the owner of the card,

[8] For the sake of clarity, we assume that b can be represented as a binary vector of length n (i.e., $b \in \{0,1\}^n$). In practice, this might not be true when the matching of templates relies on geometric information (for instance in fingerprints), in which case the error-correcting approach has to be adapted to this situation.

[9] In the basic version l, the length of the random string generated, is smaller than n, the length of the biometric profile. However, this is not really a problem as it is possible to use $rand$ as a seed of a good pseudorandom number generator to generate an almost uniformly random string of arbitrary size.

it applies the Retrieve operation on a fresh biometric sample b' measured by the biometric sensor which outputs $rand'$ and accepts the current user as the rightful owner of the card only if $h(rand) = h(rand')$.

Another application of fuzzy extractors is the possibility of using the biometric input of the user as a key to encrypt and authenticate the user's data. For instance, $rand$ can act as an encryption key which can be retrieved only by the combination of the user's biometric profile and the helper string. As $rand$ is never explicitly stored and the user's biometrics acts as a key, this guarantees that only if the correct biometric template is presented, the record of the user can be decrypted.

4 Operation and Use of the Privacy-Preserving Identity Card

We suppose that the privacy-preserving identity card is a contact smartcard that has sufficient resistance against physical and logical attacks [10] (see Section 3.1). The smartcard contains a processor that can compute efficiently cryptographic primitives such as asymmetric encryption and group signature verification. The card memory stores identity data similar to those printed on existing identity cards (e.g., names, date and location of birth, address, etc.), plus biometric data and other security-related information, such as public and private keys.

When the smartcard is inserted into a reader device, the smartcard processor initiates a *mutual authentication* between the card and the reader (see Section 5.2 for more details). If the mutual authentication fails, the smartcard is not activated (i.e., its processor does nothing). Contrarily, when the mutual authentication succeeds, the embedded processor initiates a *biometric verification* of the user, by using for instance the fuzzy commitment scheme for biometric authentication described in Section 3.3. Finally, when the biometric authentication is successful, the processor initiates a *question-response* protocol with the reader device. In pratice, the question of the reader could be any binary query related to an attribute of the user such as "Is the user a Finnish citizen?" (for instance when crossing the border), "Is the user under 18 years old?" (when proving that the user is within some age interval), "Is the user firstname Alice?" (when checking the identity before boarding a plane) or "Is the user an inhabitant of Toulouse?" (when accessing a local service restricted to municipality residents).

If the question-response protocol is implemented through an anonymous credential system that is expressive enough to prove any combination of the logical operations AND, OR and NOT regarding the attributes of the user then it is possible in principle to check any particular binary statement regarding his identity[11]. Note that in any case, the card discloses no personal data, only a binary statement on data provided by the reader, i.e., data that already exist out of the

[10] We also assume that the smartcard reader device that will interact with the privacy-preserving identity card possesses similar tamper-proof properties.

[11] See for instance [8] for an efficient implementation of anonymous credentials that allows to prove AND, OR and NOT statements regarding the attributes encoded.

card. For instance, for checking the first name Alice, this information must be sent by the reader to the card, either because the user has claimed it, or because it has been read on another document such as a boarding pass.

5 Implementations of the Privacy-Preserving Identity Card

The two implementations of the privacy-preserving identity card that we propose combine the different technologies and concepts briefly reviewed in Section 3. We call the first implementation BasicPIC, which stands for Basic implementation of a Privacy-preserving Identity Card (PIC). In this implementation, we suppose that the smartcard tamperproofness is "sufficient". In practice however, it is quite likely that if an adversary spends enough resources and time he will be able to break the tamper-proofness characteristic of the smartcard and read and/or modify the information stored on it. If this assumption is broken, for instance if the adversary is able to access the memory of the smartcard, this can greatly endanger security properties such as *no personal information leakage*, *authenticity*, *unforgeability* and *correctness*. To overcome this limitation, we propose an extended implementation of the privacy-preserving identity card that we call ExtendedPIC. The main idea of this implementation is to complement the functionalities of BasicPIC with the use of *fuzzy extractors* to protect the information stored in the card and *non-interactive zero-knowledge proofs* as a privacy-preserving proof of statements related to the user's data.

5.1 Initialisation

When the user wishes to acquire a new privacy-preserving identity card, he goes to an Registration Authority (RA) who can verify the personal data of the user and register the demand. We denote by $a_1, \ldots, a_k$, the k attributes of the user that embodies his identity. For instance, the i^{th} attribute a_i could be a name (string of characters value), a year of birth (integer value) or an address (mix of strings of characters and integers). After having checked the identity of the user, the RA scans a biometric profile of the user b (which could be for instance his fingerprints, a map of his iris or a sample of his voice). The RA sends b in a secure manner along with the personal information of the user to the Certification Authority (CA). The secure transmission of the personal information of the user between the RA and the CA is done by communicating over an electronic secure channel or via a physical delivery whose process is under strict monitoring.

The CA is responsible for issuing the privacy-preserving identity card and for signing the user's information in order to produce the anonymous credentials. The CA also performs the Join operation (see Section 3.2) to generate the signing key SKG_U of the user for the group signature. This key is stored within the tamper-proof smartcard that is the core of the privacy-preserving identity card. For an external observer, the card is "blank" and looks exactly the same as any other privacy-preserving identity card. The exact form of the smartcard can vary, depending on the chosen trade-off between the individual cost of each card that we are willing to spend and the assumptions we make on the time and means that the adversary is able to deploy. If the technology is affordable, the card could possess a biometric sensor[12] and a screen. The screen could display for instance the identifier of the reader and the questions asked to the card.

In ExtendedPIC, the credentials emitted by the CA take the form of the CA's signature on the attributes of the user. Specifically, we denote these credentials by $\sigma_{CA}(a_1), \ldots, \sigma_{CA}(a_k)$, where $\sigma_{CA}(a_i)$ is the signature on the i^{th} attribute of the user using the CA secret key. The operation of the fuzzy extractor Generate is performed on the biometric profile of the user b and produces as output a random string $rand$ and an helper string p. The random string $rand$ will be used as the key to encrypt[13] the attributes of the user, $a_1, \ldots, a_k$, and the signatures of the CA on these attributes $\sigma_{CA}(a_1), \ldots, \sigma_{CA}(a_k)$. The attributes and their associated signatures are stored encrypted inside the card but the helper string p can be stored unprotected.

Although it would be also possible to store an encrypted version of the signing key of the user SKG_U we suppose for the sake of clarity that it is not the case and that the mutual authenticity checking as well as the biometric verification are performed in the same manner in BasicPIC and ExtendedPIC. In practice however, SKG_U could also be encrypted using the key extracted from the fuzzy extractor, which requires that the biometric profile of the user is acquired first during the Retrieve operation in order for the mutual authenticity protocol to succeed. In this situation, it is possible to combine in a natural manner the biometric verification and the mutual authenticity checking into a single protocol. This protocol would fail if the biometric profile acquired during the Retrieve operation does not correspond to that of the valid owner of the card or if the card does not possess a valid private signature key SKG_U.

Before an organization can use a reader device able to interact with privacy-preserving identity cards, the organization needs first to register the device to the CA. The CA then emits a credential cr in the form of "This reader is allowed to ask the question f to a privacy-preserving identity card. The answer to this question has to be encrypted using the public encryption key EK_R.". The public encryption key EK_R is supposed to be specific to the reader and as such can be considered as its identifier. The CA will certify this credential by performing Sign(cr, SK_{CA}) which generates $\sigma_{CA}(cr)$, the signature on the credential cr using the CA secret key. The reader also knows the group verification key VKG which is public and will be used to check the authenticity of a privacy-preserving identity card during the group signature.

[12] Some companies, such as Novacard, have started to sell smartcard integrating a fingerprint sensor directly on the card since at least 2004. If the privacy-preserving card is integrated within the cell-phone of the user, it is also possible to imagine that iris recognition could be easily implemented if the cell-phone possesses a camera.

[13] For example, the encryption scheme used can be a symmetric scheme where $rand$ acts as the key for encrypting and decrypting data. For instance l, the size in bits of $rand$ can be set to be the size of an AES key (128 or 256 bits).

5.2 Mutual Authenticity Checking

Before the card answers questions of a particular reader, it needs to ensure that 1) the reader is an authentic device and 2) it possesses the corresponding credentials. On the other hand, the reader has to check that the card is a genuine privacy-preserving identity card but without learning any information related to an identifier of the card or its user. Regarding the scheme used for signing the credential, any standard signature scheme such as DSA or ECDSA can be used to implement this functionality in practice. The mutual authenticity checking protocol consists in three rounds of communication:

1. During the first round, the card generates dynamically a new pair of encryption/decryption keys (EK_{temp}, DK_{temp}). The public encryption key EK_{temp} will be used by the reader to encrypt the information it sends to the card during this session while the decryption key DK_{temp} is kept secret in the card. The card also computes $\sigma_{G,U}(EK_{temp}) \leftarrow \mathsf{SignGroup}(EK_{temp}, SKG_U)$, which corresponds to a group signature on the encryption key EK_{temp}. The card sends in clear EK_{temp} and $\sigma_{G,U}(EK_{temp})$ to the reader. The reader considers the group signature as valid (and proceeds to the second round) if $\mathsf{VerifySignGroup}(EK_{temp}, \sigma_{G,U}(EK_{temp}), VKG)$ outputs accept or aborts the protocol otherwise.
2. During the second round, the reader uses the card's public key EK_{temp} to encrypt its credential cr, the signature of the CA on this credential $\sigma_{CA}(cr)$ as well as a randomly generated string of bits r, and sends this encrypted message to the card. The card performs $\mathsf{VerifySign}(cr, \sigma_{CA}(cr), VK_{CA})$ and either accepts the reader and goes to the third round, or rejects and aborts the protocol. The card should have a built-in mechanism that limits the number of attempts that a reader may try within some time window.
3. During the third round, the card generates a random nonce x and computes $\sigma_{G,U}(r||x) \leftarrow \mathsf{SignGroup}(r||x, SKG_U)$, which corresponds to a group signature on the concatenation of the random string of bits r and x. Afterwards, the card sends to the reader the cipher $ciph \leftarrow \mathsf{Encrypt}(x||\sigma_{G,U}(r||x), EK_R)$, where $ciph$ corresponds to the encryption of the message $x||\sigma_{G,U}(r||x)$ with the readers public key EK_R. Finally, the reader decrypts this message by performing $\mathsf{Decrypt}(ciph, DK_R)$ which reveals $x||\sigma_{G,U}(r||x)$. The reader recognizes the card has a genuine one only if $\mathsf{VerifySignGroup}(r||x, \sigma_{G,U}(r||x), VKG)$ has for outcome accept. Otherwise, the reader aborts the protocol.

Suppose that the reader stores in a list all the pairs of random strings/nonces and group signatures $(r||x, \sigma_{G,U}(r||x))$ that he has seen along with other information such as a time stamp. As such, this list is of no use for it to break the privacy of users as it is not even able to recognize if two different signatures belong to the same individual or not. However in some extreme situation where there is a clear necessity of lifting the anonymity of a particular signature, the reader may hand over the pair $(r||x, \sigma_{G,U}(r||x))$ to the CA which will be able to retrieve SKG_U by performing $\mathsf{LiftAnonymity}(r||x, \sigma_G(r||x))$ and thus also the identity of U.

5.3 Biometric Verification

The privacy-preserving identity card is activated by the verification of the biometrics of its user. During this phase, a fresh biometric sample b' of the user is acquired by the biometric sensor and sent to the card which then performs the $\mathsf{Retrieve}$ operation upon it. This operation consists in computing $rand'$ using the fuzzy extractor together with the helper string p and calculating $h(rand')$. The outcome of this procedure is either accept or reject depending on whether or not $h(rand') = h(rand)$. If the user passes the verification test, the card is considered activated and enters the question-response protocol. Otherwise, the card refuses to answer external communication.

5.4 Privacy-Preserving Proof of Statements

Basic Implementation. Let $f(a_i)$ be the binary answer to a boolean question f about the attribute a_i of the user (or a combination of attributes). For instance, the semantic of the bit $f(a_i)$ could be true if its value is 1 and false if its value is 0. The question f as well as the public encryption key EK_R of the reader have been transmitted as part of the credential cr. First, the card concatenates the answer bit $f(a_i)$ with the random string r sent by the reader during the mutual authentication phase to obtain $f(a_i)||r$ and signs it, which generates $\sigma_{G,U}(f(a_i)||r)$. The card computes the cipher $ciph \leftarrow \mathsf{Encrypt}(f(a_i)||r||\sigma_{G,U}(f(a_i)||r), EK_R)$, where $ciph$ corresponds to the encryption of the message $f(a_i)||r||\sigma_{G,U}(f(a_i)||r)$ with the readers public key EK_R. Afterwards, the reader decrypts this message by performing $\mathsf{Decrypt}(ciph, DK_R)$ which reveals $f(a_i)||r$ and $\sigma_{G,U}(f(a_i)||r)$. The reader first verifies the validity of the signature $\sigma_{G,U}(f(a_i)||r)$ with the verification key of the group and trusts the answer $f(a_i)$ only if this verification succeeds. Note that in the implementation $\mathsf{BasicPIC}$, the correctness of answer $f(a_i)$ relies partly on the assumption that the card is tamperproof and therefore cannot be made to misbehave and lie to a question asked by the reader.

Consider an adversary that would like to play a relay attack by transmitting the communication normally between a genuine card and a genuine reader during the mutual authentication phase and then hijacks the session during the question-response protocol by acting as the card. If the answer bit was not signed with the private signature key of the card, the adversary could set the answer to the reader's question to his own choice. Moreover, the encryption scheme used has to be *semantically secure*[14] in order to avoid the possibility of an adversary having an advantage in guessing whether the answer of the card to the reader's question is 0 or 1. As a semantically secure encryption is necessary also probabilistic, this ensures that even if the card answers twice to the same question it will not be possible for an eavesdropper to distinguish whether these

[14] Ideally, the encryption scheme should even fulfill a stronger security requirement called *indistinguishability under adaptive chosen ciphertext attack* (IND-CCA2) (see [22] for instance). This property has been proven to also guarantee the *non-malleability* property and thus the threat of an adversary flipping the bit of the answer transmitted.

two answers where produced by the same privacy-preserving identity card or two different cards. In practice, this encryption scheme could be for instance the Cramer-Shoup cryptosystem [11] which has been one of the first proven to satisfy the IND-CAA2 property.

Extended Implementation. In our setting, the card wants to prove to the reader some function related to the attributes of the user and also that these attributes have been signed (certified) by the CA. However in ExtendedPIC, since we relax the tamper-proofness requirement we want to go beyond simply sending a signed answer bit, by issuing a zero-knowledge proof. This can be done as follows:

1. We suppose that the binary question asked by the reader is related to the i^{th} attribute of the user. The card performs Retrieve by taking as input a fresh biometric sample of the user b' and the helper string p stored on the card. The output of the Retrieve operation is the random string $rand$ which is used as a key to decrypt the values of the attribute a_i and its associated signature $\sigma_{CA}(a_i)$ from their encrypted versions stored on the card.
2. The card computes $comm(a_i) \leftarrow \mathsf{Commit}(a_i, aux)$, where $comm(a_i)$ is a commitment on the value of the i^{th} attribute a_i and aux is some auxiliary information needed to open the commitment. In practice, we propose to use the Groth-Sahai commitment scheme [17], which is perfectly binding (thus forbidding that the card can change afterwards the value of the attribute committed and therefore prove a false statement) and computationally hiding (thus preventing a reader to learn the value of the attribute committed unless he can break some computational assumption).
3. The card computes $\pi \leftarrow \mathsf{Prove}((a_i, \sigma_{CA}(a_i), aux)|\mathsf{VerifySign}(a_i, \sigma_{CA}(a_i), VK_{CA}) = \mathsf{accept} \wedge a_i = \mathsf{Open}(comm(a_i), aux) \wedge f(a_i) = true)$, where VK_{CA} is the public verification key of the CA that can be used to check the validity of the CA's signature, $\sigma(a_i)$ is the signature by the CA of attribute a_i and $f(a_i)$ is a boolean question regarding a_i. Effectively, π is a non-interactive zero-knowledge proof of the following statement "The user of this privacy-preserving identity card knows how to open the commitment $comm$ to some value a_i, and this value has been signed by the CA, and when the boolean function f is computed on a_i it returns true" which could be summarized as "The CA certifies that the user of this privacy-preserving identity card satisfies the boolean question f when it is applied on his i^{th} attribute". The boolean question f could be any binary property related to an attribute of the user. The idea of using a zero-knowledge proof can also be extended so as to prove a binary statement regarding several attributes at the same time, such as a conjunction.
4. The card sends $\mathsf{Encrypt}(comm||\pi, EK_R)$ to the reader which then decrypts it and verifies the validity of the proof.

For the practical implementation of the privacy-preserving proof of statements, we suggest to use the recent non-interactive zero-knowledge proofs developed by Belenkiy, Chase, Kohlweiss and Lysyanskaya [2]. These proofs are an extension of the CL-signatures [7] and have been proven secure on the common reference string model. These non-interactive zero-knowledge proofs are based partly on the Groth-Sahai commitment scheme [17] that has some interesting non-trivial properties such as being *f-extractable*, which means it is possible to prove that the committed value satisfies a certain property without revealing the value itself, and allows *randomizability*, which means that a fresh independent proof π' of the same statement related to the committed value can be issued from a previous proof π of this statement. In the context of the privacy-preserving identity card, the f-extractability property allows to show that an attribute of the user satisfies some binary property without disclosing the attribute itself whereas the randomizability property ensures that even if the card prove several times the same statement, the reader will see each time a different proof of this statement, thus avoiding the risk of linkability between them.

5.5 Analysis of the Implementations

The implementations BasicPIC and ExtendedPIC fulfill the desiderata of a privacy-preserving identity card (as listed in Section 2) as they respect the following properties:

- *No personal information leakage*: in BasicPIC, due its tamper-proof characteristics, the attributes describing the user are safely stored on the smartcard and only one bit of information regarding the user is revealed every time the card answers a question. In ExtendedPIC, the attributes of the user are stored in the smartcard encrypted and can only be decrypted if the user biometric profile is presented as input to the fuzzy extractor in conjunction with the helper string. Moreover, the card answers to a question of the card by showing a non-interactive zero knowledge proof which leaks nothing but one bit of information about the validity of a particular binary statement.
- *Unlinkability*: the use of a group signature prevents the possibility of linking the proofs of two different statements to the same user. Moreover, there is no such thing as a pseudonym or an identifier used in our implementations (with exception of the group signing key SKG_U, which is never disclosed by the card). In particular, there is no identity card number, which could be used to trace all the card uses and the public key for the session EK_{temp} is generated dynamically at random by the card and has no link with its identity. In ExtendedPIC, the randomizability property of the non-interactive zero-knowledge proof also ensures that even if the card proves several times the same statement, the proofs generated will be different and look as if they were independent.
- *Ownership proof*: before its activation, the card will check that the current user is effectively the legitimate owner of the card by verifying his biometrics. In ExtendedPIC, the biometric template of the user is also used as input to the fuzzy extractor when it is time to decrypt the data stored on the card during the privacy-preserving proof of statements.
- *Authenticity*: the reader will prove its authenticity and its right to ask a particular question by showing the corresponding credential signed by the

CA. The card will prove its authenticity by showing that it can sign a randomly generated message on the behalf of the group of genuine privacy-preserving identity card, and also indirectly in ExtendedPIC by showing the non-interactive zero-knowledge proof that it possesses the signature of CA on the attributes of the user.

- *Correctness*: in BasicPIC, the correctness of a statement proven by the card relies mainly on the fact that the tamper-proof properties of the smartcard forbids a dishonest user from changing its designed behaviour. Indeed, the card can be seen as a kind of oracle that never lies to a question asked to it. To change the behaviour of the oracle would require breaking the smartcard, which would violate the tamper-proof assumption. Moreover as the answer is encrypted using a non-malleable asymmetric encryption scheme using the public key of the reader, it is impossible for a potential adversary to flip the answer bit without being detected. Finally, as the answer bit is signed with the key of the card, this prevents an adversary from impersonating a valid card during the question-response protocol. In ExtendedPIC, the correctness of a statement proven by the card is a direct consequence of the soundness and completeness properties of the non-interactive zero-knowledge proof used.
- *Unforgeability*: this property is ensured by the tamper-proofness of the smartcard, as well as the fact that the data of the user are stored encrypted on the card, plus by the verification of the credential issued by the CA and the signatures of the CA on the attributes of the user.
- *Unforgeability*: this property is ensured by the tamper-proofness of the smartcard, as well as the fact that the data of the user is stored encrypted on the card, plus by the verification of the credential issued by the CA and the signatures of the CA on the attributes of the user.
- *Optional anonymity removing*:in extreme situations, the anonymity of the actions of the user of a privacy-preserving identity card can be lifted by having the CA cooperating with a verifier and applying the LiftAnonymity operation on the corresponding pair of random string and associated signature.
- *Explicit consent*: in most situations, the user expresses his consent by inserting his card in the reader: we can consider that, since the reader has to be certified by the CA, it is trustworthy enough, i.e., tamper-proof and able to display correctly the question on a screen (part of the reader). Then if the user accepts the question, he just confirms it by pushing a switch, else he just withdraws his card from the reader. If the reader cannot be trusted and if the question can be too sensitive, the card should be equipped with embedded screen and switch.

6 Possible Extensions and Conclusion

Potential applications of the privacy-preserving identity card may include access to online services such as e-government services and e-business applications. In this context, the card could be plugged into a standard personal computer via an external trusted USB reader certified and sold by the government. In this case, all the communication between the card and the e-government platform hosting the online services should be encrypted to prevent potential information leakage, e.g. to a spyware that would have infected the user's personal computer. Of course, in this virtual context it may be more difficult for a user to keep an explicit control on how his data are used and we may have to cope with more threats than in the simple card-reader interaction scenario. Another possible extension to the privacy-preserving identity card is to embed it directly in a device such as a cellular phone which raises the question of how much trust can be put in such a device.

Such extensions would require an in-depth security analysis to ensure that they can be safely integrated in a privacy-preserving identity card. But with the basic and extended implementations that we have described, it is technically feasible to develop and deploy a privacy-preserving identity card with currently available technologies. Whether governments and law enforcement authorities would accept such a card to be deployed is another question.

Acknowledgments

We would like to thanks the anonymous experts who reviewed a previous version of the paper for their insightful comments which have helped us to improve the quality of this paper.

References

1. Bangerter, E., Camenisch, J., Lysyanskaya, A.: A cryptographic framework for the controlled release of certified data. In: Proceedings of the 12th International Security Protocols Workshop, pp. 20–42 (2004)
2. Belenkiy, M., Chase, M., Kolhweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008)
3. Birch, D.: Psychic ID: A blueprint for a modern national identity scheme. Identity in the Information Society 1(1) (2009)
4. Boudot, F.: Partial revelation of certified identity. In: Proceedings of the First International Conference on Smart Card Research and Advanced Applications (CARDIS 2000), pp. 257–272 (2000)
5. Calmels, B., Canard, S., Girault, M., Sibert, H.: Low-cost cryptography for privacy in RFID systems. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 237–251. Springer, Heidelberg (2006)
6. Camenisch, J., Lysyanska, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
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., Thomas, G.: Efficient attributes for anonymous credentials. In: Proceedings of the 2008 ACM Conference on Computer and Communications Security (CCS 2008), pp. 345–356 (2008)
9. Chaum, D.: Security without identification: transaction systems to make Big Brother obsolete. Communications of the ACM 28(10), 1030–1044 (1985)
10. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
11. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
12. Deswarte, Y., Gambs, S.: Towards a privacy-preserving national identity card. LAAS Report No.09208, 20 pages (August 2009), http://hal.archives-ouvertes.fr/hal-00411838/fr/
13. Dodis, Y., Reyzin, L., Smith, A.: Security with Noisy Data. In: Tuyls, P., Skoric, B., Kevenaar, T. (eds.) Fuzzy extractors, a brief survey of results from 2004 to 2006, ch. 5. Springer, Heidelberg (2007)
14. European Union, Directive 95/46/EC of the European Parliament and of the Council of 24 October (1995), on the protection of individuals with regard to the processing of personal data and on the free movement of such data, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML
15. European Network and Information Security Agency (ENISA) position paper, Privacy features of European eID card specifications, http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_privacy_features_eID.pdf
16. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(3), 691–729 (1991)
17. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
18. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS 1999), pp. 28–36 (1999)
19. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems (Extended abstract). In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 184–199. Springer, Heidelberg (2000)
20. PRIME - Privacy and Identity Management for Europe, PRIME white paper (May 2008), https://www.prime-project.eu/prime_products/whitepaper/
21. Ravi, S., Raghuanathan, A., Chadrakar, S.: Tamper resistance mechanisms for secure embedded systems. In: Proceedings of the 17th International Conference on VLSI Design (VLSID 2004), pp. 605–611 (2004)
22. Shoup, V.: Why chosen ciphertext security matters. IBM Research Report RZ 3076 (November 1998)

Using SAT-Solvers to Compute Inference-Proof Database Instances

Cornelia Tadros and Lena Wiese

Technische Universität Dortmund, 44221 Dortmund, Germany
{tadros,wiese}@ls6.cs.uni-dortmund.de
http://ls6-www.cs.tu-dortmund.de/issi/

Abstract. An inference-proof database instance is a published, secure view of an input instance containing secret information with respect to a security policy and a user profile. In this paper, we show how the problem of generating an inference-proof database instance can be represented by the partial maximum satisfiability problem. We present a prototypical implementation that relies on highly efficient SAT-solving technology and study its performance in a number of test cases.

1 Introduction and System Settings

Controlled Query Evaluation (CQE) is a framework for inference control in logical database systems. In [3] a preprocessing procedure (which we call *pre*CQE here) is described that accepts propositional input (an input instance, a confidentiality policy, an availability policy and a user profile). It outputs an "inference-proof" solution instance; this output instance is secure in the sense that it can be published to provide answers to any user queries without enabling the user to deduce any confidential information – and without the need to maintain a history of previous user queries. As secondary and tertiary goals, the output instance is meant to preserve maximum availability (of entries in the availability policy) as well as minimize the amount of modifications ("distortions") with respect to the input instance. The aims of this article are twofold:

1. We show that precomputing an inference-proof, availability-preserving, and distortion-minimal database instance can be reduced to a weighted partial MAXSAT (W-PMSAT) problem with three weights.
2. We present and evaluate a prototypical implementation where highly efficient third-party SAT solving tools can be plugged in – instead of implementing the algorithm (as theoretically exposed in [3]) directly.

Our preprocessing approach stands orthogonal to history-based inference control mechanisms in logical databases (as for example in [2,11]) that compute the (possibly distorted) answers at runtime. Yet, it is akin to the use of cover stories (see [8,5]) in multilevel secure databases while adding the bonuses of availability preservation and distortion minimization.

We now describe components (visualized in Figure 1) and settings of the CQE system that are assumed in this article. The system is based on a propositional

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 65–77, 2010.

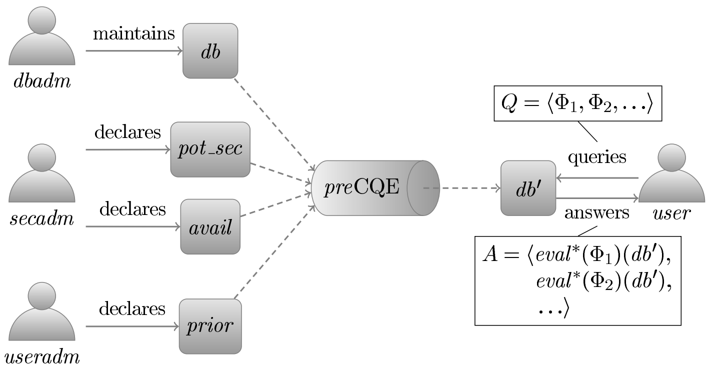

maintains
db
dbadm
declares
pot_sec
secadm
declares
avail
declares
prior
useradm
preCQE
db′
$Q = \langle \Phi_1, \Phi_2, \ldots \rangle$
queries
answers
user
$A = \langle eval^*(\Phi_1)(db'), eval^*(\Phi_2)(db'), \ldots \rangle$

Fig. 1. Concept of the algorithm

language with an infinite number of propositional variables (the propositional "alphabet" $\mathcal{P}$). For our running example, $\mathcal{P}$ is the vocabulary for a medical record with diseases and medications:

$$\mathcal{P} = \{\texttt{cancer}, \texttt{aids}, \texttt{flu}, \texttt{cough}, \ldots, \texttt{medA}, \texttt{medB}, \texttt{medC}, \ldots\}$$

Propositional formulas are built from $\mathcal{P}$ with the connectives $\wedge$, $\vee$ and $\neg$. A propositional variable is also called a "positive literal"; a propositional variable preceded by a negation sign is called a "negative literal".

Data Model. The input database instance *db* is a finite set of propositional variables (where each variable represents a tuple in the database); hence $db \subset \mathcal{P}$. It represents a complete interpretation I^{db} for all variables in $\mathcal{P}$: a variable $A \in db$ is interpreted as *true*, otherwise it is interpreted as *false*. In our example, $db = \{\texttt{cancer}, \texttt{aids}, \texttt{medA}, \texttt{medB}\}$ comprises the set of all *true* propositions, while all other variables (from $\mathcal{P} \setminus db$) are *false*. The input instance is maintained by the database administrator *dbadm*.

Interaction Model. A user is assumed to interact with a database instance via an evaluation function $eval^*$; it takes a query formula and a database instance as inputs and returns the query formula or its negation depending on which of the two formulas is true in the instance:

$$eval^*(\Phi)(db) = \begin{cases} \Phi & \text{if } I^{db} \models \Phi \quad \text{(with } \models \text{ being the model operator)} \\ \neg\Phi & \text{else} \end{cases}$$

Eg., $eval^*(\texttt{flu})(db) = \neg\texttt{flu}$ and $eval^*(\texttt{medB} \wedge \texttt{medA})(db) = \texttt{medB} \wedge \texttt{medA}$.

Confidentiality Model. The confidentiality policy *pot_sec* is a finite set of formulas. An entry of *pot_sec* is a "potential secret": the user must not know that a potential secret is *true* in *db*, but he may assume that it is *false*. The confidentiality policy is declared by the security administrator *secadm*, for example as $pot\_sec = \{\texttt{cancer}, \texttt{aids}\}$: the user may not know the fact `cancer` (or `aids`), but he may learn $\neg$`cancer` (and $\neg$`aids`). As a rational, sophisticated person, the user is assumed to know the policy specification *pot_sec*. The only protection mechanism analyzed in this article is modification of some *db*-entries; hence a solution instance may contain the entry `flu`. This is called "uniform lying" in the CQE context.

Availability Model. The availability policy *avail* is a finite set of formulas. It specifies important information, that should at best not be distorted by the lying mechanism. That is, whenever it is possible to distort information not contained in *avail* (while still protecting the secrets), we prefer this distortion to a distortion affecting *avail* entries. The availability policy is also declared by the security administrator *secadm*, for example as $avail = \{\texttt{medA} \wedge \texttt{medB}, \texttt{medB}\}$ stating that the information whether *both* `medA` and `medB` or `medB` *alone* are prescribed should not be distorted (due to side effects or mutual reactions with substances that must be considered). Beyond this explicit goal to preserve availability, there is a tertiary goal to distort as few database entries as possible.

User Model. The user profile *prior* is a finite set of formulas containing a specification of the knowledge the user had prior to interacting with the CQE system. The user profile is declared by the user administrator *useradm*; eg., the user knows that a patient taking medicine A is ill with Aids or Cancer, and a patient taking medicine B is ill with Cancer or Flu: $prior = \{\neg\texttt{medA} \vee \texttt{cancer} \vee \texttt{aids}, \neg\texttt{medB} \vee \texttt{cancer} \vee \texttt{flu}\}$. He is able to use full implication on his knowledge and the database answers to deduce other facts from them. Hence, the user knowledge and the database answer must never be inconsistent as from inconsistent knowledge the user can deduce any facts (including the secrets): from a contradiction, anything follows by logical implication. Beyond the mere representation of the user knowledge in the user profile, the user is assumed to be aware of the system settings (that is, complete database, known policy, lying).

Execution Model. The *pre*CQE procedure takes *db*, *pot_sec*, *avail* and *prior* as inputs and outputs a complete instance db'. The output instance db' has the property that it is consistent with the a priori knowledge *prior* and that no truthful answer to any user query enables the user to infer a potential secret from *pot_sec*. More formally, in the case of a complete db' and lying, we define "inference-proofness" of db' as follows:

Definition 1 (Inference-proofness). *A complete database instance db' is called* inference-proof *(with respect to prior and pot_sec) iff*

1. $I^{db'} \models prior$
2. $I^{db'} \not\models \Psi$ *for every* $\Psi \in pot\_sec$

A user (modeled by *prior*) can pose any query sequence $Q = \langle \Phi_1, \Phi_2, \dots \rangle$ and retrieve truthful responses $A = \langle eval^*(\Phi_1)(db'), eval^*(\Phi_2)(db'), \dots \rangle$ from an inference-proof instance without being able to deduce a secret. Eg., neither $db'_1 = \{\texttt{medB}, \texttt{flu}\}$ nor $db'_2 = \emptyset$ disclose any of the secrets `aids` and `cancer` but they obey *prior* – hence both are inference-proof.

In [3] it is shown that with these system settings, the problem of finding an inference-proof instance db' amounts to finding a model (a satisfying interpretation) $I^{db'}$ for a constraint set C. This set C consists of the user profile and the negations of the potential secrets (a condition for consistency of C – and hence existence of a db'– is identified in [3]):

Definition 2 (Constraint set). *For a set prior and a set pot_sec, the* constraint set *is*

$$C := prior \cup Neg(pot\_sec) \qquad where\ Neg(pot\_sec) := \{\neg\Psi \mid \Psi \in pot\_sec\}$$

Hence, in our example the constraint set C is:

$$C := \{\neg\mathtt{medA} \vee \mathtt{cancer} \vee \mathtt{aids}, \neg\mathtt{medB} \vee \mathtt{cancer} \vee \mathtt{flu}, \neg\mathtt{cancer}, \neg\mathtt{aids}\}$$

and it holds that $I^{db'_1} \models C$ as well as $I^{db'_2} \models C$.

To meet the availability requirements and thus retain as much correct information in db' as possible, we define two distance measures: the first one to measure how many *avail* entries are affected by distortion and the second one to measure how many *db* entries are affected by distortion:

Definition 3 (Availability preservation/distortion minimization). *The* availability distance *(for inference-proof db') is defined as*

$$avail\_dist(db') := ||\{\Theta \in avail \mid eval^*(\Theta)(db') \neq eval^*(\Theta)(db)\}||$$

An inference-proof db' is availability-preserving *iff there is no db'' such that $avail\_dist(db') > avail\_dist(db'')$.*

The distortion distance *(for inf.-proof and availability-preserving db') is*

$$db\_dist(db') := ||\{A \in \mathcal{P} \mid eval^*(A)(db') \neq eval^*(A)(db)\}||$$

An inference-proof and availability-preserving db' is distortion-minimal *iff there is no db'' such that $db\_dist(db') > db\_dist(db'')$.*

We first of all minimize *avail_dist* and among the *avail_dist*-minimal solutions search for one that minimizes *db_dist*. Yet, due to the model requirement, inference-proofness and hence confidentiality of the secrets is our main goal and the two distances are availability optimization functions. In our example, we see that db'_1 preserves availability better than db'_2: while in the input instance *db* both entries of *avail* are *true*, in db'_1 the first entry is *false* but the second is *true*, such that $avail\_dist(db'_1) = 1$; in db'_2 both entries are *false*, such that $avail\_dist(db'_2) = 2$. Hence, db'_1 is our unique optimal solution (and distortion minimality has no effect).

A crucial point for the efficiency of the *pre*CQE algorithm is that only a finite subset of the infinite $\mathcal{P}$ of "decision variables" that are contained in *prior*, *pot_sec* or *avail* have to be considered when searching for an inference-proof, availability-preserving and distortion-minimal solution:

Definition 4 (Decision variables). *The* decision variables *are*

$$\mathcal{P}_{decision} := \{A \in \mathcal{P} \mid A\ occurs\ in\ prior, pot\_sec\ or\ avail\}$$

In our example, $\mathcal{P}_{decision} = \{\mathtt{cancer}, \mathtt{aids}, \mathtt{flu}, \mathtt{medA}, \mathtt{medB}\}$.

2 Encoding as SAT Problem

*pre*CQE for propositional logic can be represented (by a transformation of the input constraints) as a variant of an optimization problem for the satisfiability (SAT) problem; in this case (as opposed to the Branch and Bound approach in [3]) the availability and distortion distances need not be maintained explicitly but are encoded into "weights". However, SAT solving normally refers to input formulas in conjunctive normal form (CNF) such that all *pre*CQE input formulas have to be converted into an equivalent set of "clauses" (a clause is a disjunction of literals).

In the following we present the representation of the *pre*CQE problem as a weighted partial MAXSAT (W-PMSAT) optimization problem. Here it is crucial to see the input as a set of clauses. Each clause has an associated non-negative integer as a weight. We use three weights: the highest one to account for inference-proofness (and hence confidentiality-preservation) for the so called "hard constraints", an intermediate one to account for availability preservation, and the lowest weight 1 for distortion minimization. The W-PMSAT optimization function is to maximize the sum of weights of satisfied clauses in an interpretation (or, equivalently, minimize the sum of weights of unsatisfied clauses). Hard constraints necessarily have to be satisfied; that is why the optimization is partial: the W-PMSAT solver only has to maximize the summed weight of the remaining satisfied "soft constraints". Our three weights are computed such that if all clauses with a lower weight are satisfied at the cost of *not* satisfying a clause with a higher weight, the summed total weight is lower and hence the solution is worse: this nicely encodes the fact that inference-proofness is our main, availability preservation our secondary, and distortion minimization our tertiary goal.

2.1 Clauses and Weights

The *pre*CQE inputs *db*, *avail* and the constraint set C (see Def. 2) are transformed into three sets of clauses: one set C_1 of soft constraints containing all clauses with lowest weight 1, a second set C_2 of ("auxiliary") soft constraints with an intermediate weight and a third set C_3 of hard constraints with highest weight. At first, all decision variables are transformed to soft constraints according to their evaluation in *db*. That is:

$$C_1 := eval^*(\mathcal{P}_{decision})(db) := \bigcup_{A \in \mathcal{P}_{decision}} eval^*(A)(db)$$

is the set of soft constraints that all have weight 1; in our example, $C_1 = \{\texttt{cancer}, \texttt{aids}, \neg\texttt{flu}, \texttt{medA}, \texttt{medB}\}$ (recall that $eval^*(\texttt{flu})(db) = \neg\texttt{flu}$).

Second, an intermediate weight has to be determined when considering the formulas in *avail*. Recall that the semantics of the availability policy is that only a maximal number but possibly not all of the formulas in $eval^*(avail)(db)$ can be satisfied in the solution instance db' (we use $eval^*(avail)(db)$ as an abbreviation for $\bigcup_{\Phi \in avail} eval^*(\Phi)(db)$). This optimization requirement leads to the problem of loss of structural information when transforming formulas in $\mathit{eval}^*(\mathit{avail})(\mathit{db})$ into CNF: If we take a formula Θ from $\mathit{eval}^*(\mathit{avail})(\mathit{db})$ and determine its CNF representation $\mathit{cnf}(\Theta)$ (in order to be processable by a W-PMSAT solver), all the clauses of $\mathit{cnf}(\Theta)$ have to be treated as "belonging together" when counting their weight. We can achieve this with the help of auxiliary propositional variables denoted S_Θ. The second set C_2 of auxiliary soft constraints consists exactly of the auxiliary variables: for each S_Θ, we add a clause S_Θ with weight $\mathit{card}(C_1)+1$ to C_2. In our example, the second set of auxiliary constraints with weight $\mathit{card}(C_1)+1=6$ is $C_2=\{S_{\mathtt{medA}\wedge\mathtt{medB}}, S_{\mathtt{medB}}\}$.

Next, for a formula Θ in $\mathit{eval}^*(\mathit{avail})(\mathit{db})$, $\mathit{cnf}(\Theta)$ is transformed as:

1. To each clause c of $\mathit{cnf}(\Theta)$ conjoin $\neg S_\Theta$ which gives us $c \vee \neg S_\Theta$
2. Add these augmented clauses to the constraint set C_3

Finally, for each constraint formula $\Phi \in C$, add the clauses of $\mathit{cnf}(\Phi)$ to C_3. All the clauses in the constraint set C_3 have as weight the sum of the weights of all the constraints at lower levels plus 1: $\mathit{card}(C_2)\cdot(\mathit{card}(C_1)+1)+\mathit{card}(C_1)+1$.

In our example, the set of hard constraints with weight $\mathit{card}(C_2)\cdot(\mathit{card}(C_1)+1)+\mathit{card}(C_1)+1=18$ is:

$$\begin{aligned} C_3 := \{&\neg\mathtt{medA}\vee\mathtt{cancer}\vee\mathtt{aids}, \neg\mathtt{medB}\vee\mathtt{cancer}\vee\mathtt{flu}, \neg\mathtt{cancer}, \neg\mathtt{aids},\\ &\mathtt{medA}\vee\neg S_{\mathtt{medA}\wedge\mathtt{medB}}, \mathtt{medB}\vee\neg S_{\mathtt{medA}\wedge\mathtt{medB}}, \mathtt{medB}\vee\neg S_{\mathtt{medB}}\}\end{aligned}$$

2.2 Solution Instance

We can show that a solution of this W-PMSAT input represents an inference-proof, availability-preserving and distortion-minimal propositional solution instance for the *pre*CQE input.

Proposition 1. *Let I^* be a solution of the W-PMSAT input, specified in Section 2.1, and db' the solution instance as obtained by*

$$db' := \{A \mid A \in db, A \notin \mathcal{P}_{decision}\} \cup \{A \mid A \in \mathcal{P}_{decision} \text{ with } I^* \models A\}.$$

Then db' is inference-proof, availability-preserving and distortion-minimal in the sense of Definition 1 and Definition 3.

We sketch the proof in the following: All hard constraints in C_3 must be satisfied in I^*, in particular the constraint set C from Definition 2 and thus db' is **inference-proof**. As for **availability preservation**, assume that $\tilde{db}$ is an inference-proof instance with better availability distance than db'. The interpretation $I^{\tilde{db}}$ over $\mathcal{P}$ can be extended to an interpretation $\tilde{I}$ over the variables $\mathcal{P}\cup\{S_\Theta \mid \Theta \in \mathit{avail}\}$ by setting S_Θ to true whenever $\mathit{eval}^*(\Theta)(\tilde{db}) = \mathit{eval}^*(\Theta)(db)$ and to false otherwise. (By the choice of the values of all S_Θ and the inference-proofness of $\tilde{db}$ all hard constraints C_3 are satisfied in $\tilde{I}$.) The total weight of all satisfied soft constraints $C_1 \cup C_2$ given $\tilde{I}$ is greater than the sum of weights of all satisfied clauses $S_\Theta \in C_2$, which amounts to

$$\begin{aligned}
&(card(avail) - avail\_dist(\tilde{db})) \cdot (card(\mathcal{P}_{decision}) + 1)\\
\geq\ &(card(avail) - (avail\_dist(db') - 1)) \cdot (card(\mathcal{P}_{decision}) + 1)\\
>\ &(card(avail) - avail\_dist(db')) \cdot (card(\mathcal{P}_{decision}) + 1) + card(\mathcal{P}_{decision})
\end{aligned}$$

As we can achieve at most that S_Θ is satisfied iff $cnf(\Theta)$ is satisfied, the value $(card(avail) - avail\_dist(db')) \cdot (card(\mathcal{P}_{decision}) + 1)$ is an upper bound to the sum of weights of all satisfied clauses in C_2 given I^*. Further, $card(\mathcal{P}_{decision})$ is an upper bound to the sum of weights of all satisfied clauses in C_1 given I^*. Hence, following the inequalities above, $\tilde{I}$ is more optimal then I^*, which is a contradiction to the optimality of I^*. Lastly, the sum of weights of unsatisfied clauses from C_1 is equal to db_dist, hence the instance db' is **distortion-minimal**. □

3 A *pre*CQE Implementation for Propositional Logic

In recent years, propositional SAT solving has seen a huge improvement in performance. Several highly efficient implementations take part in the yearly SAT competition (in conjunction with the SAT conference). As part of the SAT competition there also is a "MAXSAT evaluation" [6,1] that includes competition categories for W-PMSAT problems. Those SAT solvers often employ a Branch and Bound strategy for propositional input (similar to the one described in [3]) and beyond that implement highly efficient heuristics to speed up the search. While the SAT competition is already quite established, the MAXSAT evaluation has been organized just for the fourth time in 2009. This shows that the interest in efficient solving strategies for this optimization problem has come up very recently.

We wanted to apply this highly efficient W-PMSAT technology to our problem and benefit from up-to-date solver implementations instead of implementing our approach in [3] by hand; we developed a program that translates propositional *pre*CQE input formulas into a W-PMSAT instance. In particular, the program offers the following functionality:

1. It offers a graphical interface for the specification of the input (db, pot_sec, $avail$ and $prior$) and the presentation of the solution db'.
2. It transforms the specified input into a W-PMSAT instance by converting the input into CNF, creating the auxiliary constraints and computing the weights.
3. It transforms this input into the input format of the selected solver.
4. It calls the selected solver on this instance (in W-PMSAT encoding).
5. It measures the runtime of the whole computation as well as the runtime for the solver alone.
6. It transforms the solver output into the solution instance db'.

As the input format we chose the TPTP format for first-order formulas (see [10]) as we plan to extend our work to relational databases. It is a standard format for Automated Theorem Proving and is much more convenient to use than the propositional SAT solver input formats (e.g. DIMACS; see the rules of [1]): while with DIMACS variables are encoded by numbers, TPTP variables can be any user-defined strings. This is a great advantage because our administrators specify their input in TPTP. The SAT solvers we chose are all able to process the DIMACS format such that the *pre*CQE input is converted into this format by calling the external TPTP conversion library; the mapping from TPTP variables to propositional DIMACS variables is recorded on this occasion. In a separate step, *pre*CQE creates the necessary auxiliary constraints. Afterward, *pre*CQE calculates the weights of the W-PMSAT clauses and sets the weight for each clause as described above. With this step, the CQE input has been fully transformed into a W-PMSAT instance. On this instance, an external W-PMSAT solver is run to find an optimal solution; the runtime of the solver is internally recorded. *pre*CQE uses the mapping information between TPTP formulas and DIMACS variables to translate the SAT solver solution into a *pre*CQE output instance db'.

Our program has been tested with three W-PMSAT solvers:

- MiniMaxSAT (see [7])
- MAX-DPLL (as part of the SAT solver Toolbar; see [9])
- SAT4J (`http://www.sat4j.org/`)

MiniMaxSAT was run on a Linux system while we executed Toolbar on a Solaris platform. SAT4J is written purely in Java. With our system settings, MiniMaxSAT showed the best runtime performance; hence the test runs described in the upcoming section were all done with MiniMaxSAT.

3.1 Test Cases

To test our prototype we made an effort to simulate problems specific to databases. Tests were run with differently sized inputs and for every input size we tested 10 random permutations to avoid a bias caused by the input order. The runtime graphs below show the average runtime taken from all 10 instances per size as well as the deviation of the individual running times (in seconds for better readability); the runtime tables detail the number of decision variables and clauses for each input size as well as the running times in milliseconds (msec). The number of decision variables and clauses are decisive values when comparing the performance.

The first tests are a generalization of our running example: We identified 24 combinations of medicines and diseases (the “patient types”) that are consistent with the a priori knowledge *prior* and hence permitted in *db*. They are listed in Table 1. We used the abbreviations N1 to N24 to denote 24 different patient names. Then we (in the role of the *dbadm*) entered a propositional input instance *db* that contains each patient type exactly once; that is, if the *db* contains the entry[1] `‘n1_aids’`, it means that patient N1 suffers from Aids. Note that there are 66 propositional variables in the propositional *db*. Next, the potential secrets

[1] Actually, the exact TPTP syntax is `fof(r0,axiom,‘n1_aids’).`; we only state the relevant part here.

Table 1. Permissible patient types in *db*

n1_aids	n2_cancer	n3_flu	n4_aids, n4_cancer	n5_aids, n5_flu	n6_cancer, n6_flu
n7_aids, n7_cancer, n7_flu	n8_medA, n8_aids	n9_medA, n9_cancer	n10_medA, n10_aids, n10_cancer	n11_medA, n11_aids, n11_flu	n12_medA, n12_cancer, n12_flu
n13_medA, n13_aids, n13_cancer, n13_flu	n14_medB, n14_cancer	n15_medB, n15_flu	n16_medB, n16_aids, n16_cancer	n17_medB, n17_aids, n17_flu	n18_medB, n18_cancer, n18_flu
n19_medB, n19_aids, n19_cancer, n19_flu	n20_medA, n20_medB, n20_cancer	n21_medA, n21_medB, n21_aids, n21_cancer	n22_medA, n22_medB, n22_aids, n22_flu	n23_medA, n23_medB, n23_cancer, n23_flu	n24_medA, n24_medB, n24_aids, n24_cancer, n24_flu

and the a priori knowledge are entered (in the roles of *secadm* and *useradm*) in TPTP syntax for each of the 24 patient names as propositional formulas. For N1 the set *prior* contains[2] `'n1_medA'=>('n1_aids'|'n1_cancer')` and `'n1_medB'=>('n1_cancer'|'n1_flu')` and the set *pot_sec* contains `'n1_aids'` as well as `'n1_cancer'`. These entries are entered for all 24 patients; that is, we have 48 entries in *prior*, and 48 entries in *pot_sec*, too. In the first test, we did not use an explicit availability policy; that is, $avail = \emptyset$. As mentioned previously, all input is permuted at random to make tests independent of the order of input.

As for the weights, they are calculated for this example as follows: all the $24 \cdot 5 = 120$ decision variables are transformed into soft constraints receiving the weight 1. As there is no availability policy, there is no need for auxiliary constraints. All constraint formulas in C receive the weight 121. For this simplest input, a solution was found in milliseconds.

Obviously, we are interested in more meaningful results for databases with much more entries. The general idea for the expansion of our tests was to uniformly repeat the 24 patient types and test up to what number of repetitions a moderate runtime performance can be achieved. So, our first step was to repeat each patient type 10 times (each repetition with a new name) such that we have a *db* with 660 entries, *prior* with 480 entries and *pot_sec* with 480 entries; for 10 repetitions there are hence $24 \cdot 5 \cdot 10 = 1200$ decision variables. We ran tests up to 150 repetitions with 9900 *db* entries, 7200 *prior* and *pot_sec* entries each and 18000 decision variables. Figure 2 shows the results; what can be seen is that a huge amount of time is needed for the creation of the DIMACS input – this includes the creation of *Neg*(*pot_sec*) and the auxiliary constraints, the calculation and assignment of weights as well as the TPTP conversion, – whereas the MiniMaxSAT solver appears quite unimpressed by the increased size of the input. We

[2] Full TPTP syntax is `fof(r0,axiom,'n1_medA'=>('n1_aids'|'n1_cancer')).` and `fof(r1,axiom,'n1_medB'=>('n1_cancer'|'n1_flu')).`

	total runtime (msec)			solver runtime			dec.	clauses	
rep.	min	max	avg.	min	max	avg.	vars.	soft	hard
1	1832	2175	1930	178	208	184	120	120	96
25	10981	12246	11974	2214	3206	3092	3000	3000	2400
50	29333	32149	31304	4412	6360	6135	6000	6000	4800
75	58530	62026	60459	6503	9439	8991	9000	9000	7200
100	93275	101551	95792	8803	9001	8902	12000	12000	9600
125	139835	150095	142843	11000	11472	11171	15000	15000	12000
150	197389	206099	202067	13231	18253	16429	18000	18000	16800

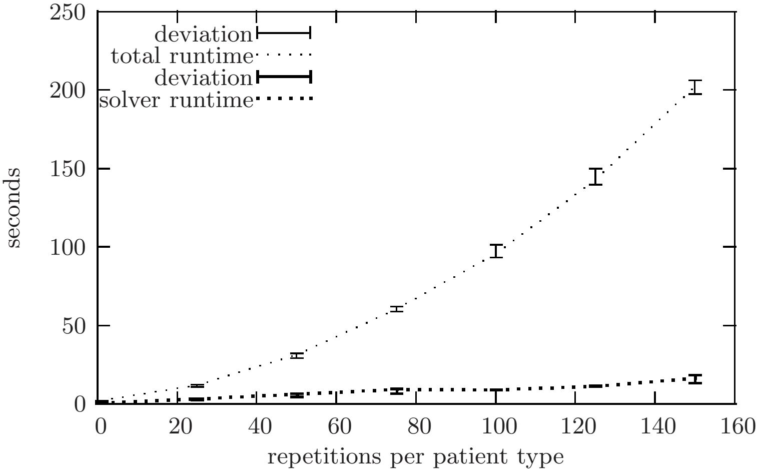


Fig. 2. Performance of *pre*CQE for 24 patient types

made two more test runs with different patient types without availability policy: a thorough analysis of the patient types reveals that there are four patient types with multiple optimal solutions. We separated them from the remaining 20 patient types with unique solution and tested the two sets separately. The existence of multiple optima slowed down the SAT solver only slightly.

After these promising results, we introduced an explicit availability policy; that is, we supplied a set *avail* with two entries for each patient type: $avail = \{\texttt{n1\_medA}, \texttt{n1\_medB}, \texttt{n2\_medA}, \texttt{n2\_medB}, \dots\}$. In the *pre*CQE implementation, they are first of all evaluated according to *db* (that is, $eval^*(avail)(db)$ is computed). As described in Section 2.1, the resulting formulas are transformed into hard constraints and auxiliary constraints with auxiliary propositional variables. In the simplest case with one repetition per patient type we thus have 120 decision variables with lowest weight 1. As there are 48 formulas in *avail*, we have 48 auxiliary constraints with weight 121. Finally there are $48 + 48 + 48 = 144$ hard constraints with weight $(48 \cdot 121) + 120 + 1 = 5929$. That is, when satisfying all hard constraints, the solution has a weight of at least 853776. We experienced problems with these high weight values, because after 55 repetitions of patient types, we faced an integer overflow: the computed solution had a negative weight. To avoid this, we then examined the performance of a reduced set of patient types. We removed the patient types with `medA`-entries, such that the first *prior* constraint will never be violated. We kept 13 patient types: N1, N2, N3, N4, N5, N6, N7, N14, N15, N16, N17, N18, N19 and their corresponding entries in *db*, *prior*, *pot_sec* and *avail*. The results can be found in Figure 3. We were able to repeat these 13 patient types much more often (up to 10150 *db* entries) than the full 24 patient type set; that is, only the search with the full set led to the integer overflow, while for the reduced set this was not the case. In comparison to tests without availability policy, runtimes increased only little (comparing the results for similar amounts of decision variables).

Lastly, we made a test with the full set of 24 patient types but we changed the potential secrets into a conjunctive format:

$$pot\_sec = \{\texttt{n1\_aids} \wedge \texttt{n1\_cancer}, \texttt{n2\_aids} \wedge \texttt{n2\_cancer}, \dots\}$$

This means that for every patient it is allowed to know if the patient has either aids or cancer but it is not allowed to know that a patient has both aids and

	total runtime (msec)			solver runtime			dec.	clauses		
rep.	min	max	avg.	min	max	avg.	vars.	low	aux.	hard
1	1744	2102	1841	142	162	146	65	65	26	78
25	8068	8308	8166	2051	2118	2077	1625	1625	650	1950
50	19028	20650	19423	4009	4121	4076	3250	3250	1300	3900
75	35061	37300	35706	5981	6266	6132	4875	4875	1950	5850
100	54295	63201	57153	5712	8375	8002	6500	6500	2600	7800
150	107971	117968	113187	8695	12533	12017	9750	9750	3900	11700
200	187700	195847	190946	11601	16924	16131	13000	13000	5200	15600
250	277757	296878	289068	15119	21247	20257	16250	16250	6500	19500
300	397551	425732	407416	18031	26073	24910	19500	19500	7800	23400
350	537890	562223	548090	22343	31778	30152	22750	22750	9100	27300

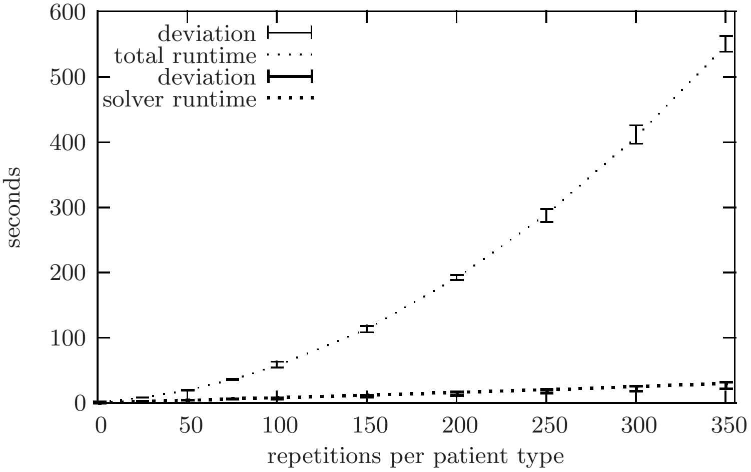


Fig. 3. Performance of *pre*CQE without `medA`-entries

	total runtime (msec)			solver runtime			dec.	clauses		
rep.	min	max	avg.	min	max	avg.	vars.	low	aux.	hard
1	1941	2934	2182	225	337	264	120	120	48	120
25	18283	23445	20439	4630	6530	5362	3000	3000	1200	3000
50	50486	58167	52449	9651	12052	9966	6000	6000	2400	6000
75	101453	105935	103266	15904	16270	16115	9000	9000	3600	9000
100	164611	175434	170920	18185	23374	22623	12000	12000	4800	12000
125	252737	276016	260537	28020	32737	31255	15000	15000	6000	15000
150	351488	380984	367471	32437	40160	39087	18000	18000	7200	18000

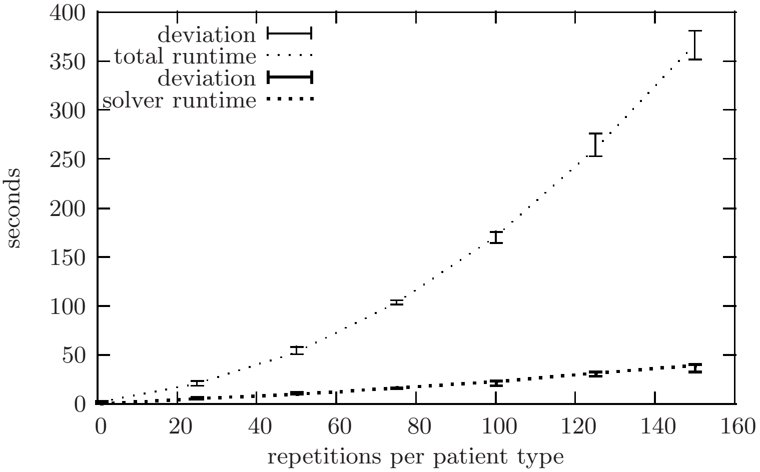


Fig. 4. Performance of *pre*CQE with conjunctive secrets

cancer at the same time. This offers a greater set of possible solutions and the SAT solver is forced to make more decision steps. Yet, as the amount of formulas in *pot_sec* is half of what it was before – only one entry per patient – the number of hard clauses is reduced: for one repetition we have 120 low level constraints, 48 auxiliary constraints and $48 + 48 + 24 = 120$ hard constraints. The results (for up to 9900 *db* entries) are detailed in Figure 4. Again, there is only a slight increase in runtime (compared with respect to the number of decision variables).

The above stated "medical record" tests contained independent subproblems in the sense that for each patient a satisfaction of the constraints could be reached without affecting the entries of other patients. Hence, as a second class of test cases we took a set of "cascading constraints" where the search for a solution requires several splitting and backtracking steps because the clauses share variables and thus are interconnected. Moreover these cascading constraints lead to test formulas with increasing length. We perceived this to be a lot more challenging task for the SAT solvers. The simplest test input with 9 decision variables was $db = \{\mathtt{c1}, \mathtt{c2}, \mathtt{c3}\}$, $pot\_sec = \{\mathtt{c3}\}$, and lastly $prior =$ {c3<=>((~v3_1|~v3_2|~v3_3)& c2), c2<=>((~v2_1|~v2_2|~v2_3)& c1)}. Without going into detail, the modification of v-variables is always suboptimal but the solver still searches on them. When comparing the runtime for 18000 decision variables, the runtime for the cascading constraints was only 2 minutes slower than for the medical records. Hence, the overall performance was still favorable.

4 Conclusion

We showed that (and how) in the CQE setting for a complete database with a confidentiality policy of potential secrets and lying as the protection mechanism, the problem of finding an inference-proof, availability-preserving and distortion-minimal database instance can be represented as a W-PMSAT problem. The presented prototype makes use of current SAT solver technology. Two classes of test cases showed that the preprocessing approach is feasible for a large number of database entries. Ongoing work at our department includes a prototypical implementation for relational database systems whose theoretical foundation is described in [4]. A major open question is whether after an update of the database instance or the policy parts of a previous solution can be reused for example with an incremental SAT solver.

References

1. Argelich, J., Li, C.M., Manyà, F., Planes, J.: MaxSAT evaluation, http://www.maxsat.udl.cat/
2. Biskup, J., Bonatti, P.A.: Controlled query evaluation with open queries for a decidable relational submodel. Annals of Mathematics and Artificial Intelligence 50(1-2), 39–77 (2007)
3. Biskup, J., Wiese, L.: Preprocessing for controlled query evaluation with availability policy. Journal of Computer Security 16(4), 477–494 (2008)
4. Biskup, J., Wiese, L.: Combining consistency and confidentiality requirements in first-order databases. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.): ISC 2009. LNCS, vol. 5735, pp. 121–134. Springer, Heidelberg (2009)
5. Cuppens, F., Gabillon, A.: Cover story management. Data & Knowledge Engineering 37(2), 177–201 (2001)
6. Heras, F., Larrosa, J., de Givry, S., Schiex, T.: 2006 and 2007 Max-SAT Evaluations: Contributed Instances. Journal on Satisfiability, Boolean Modeling and Computation 4(1), 239–250 (2008)
7. Heras, F., Larrosa, J., Oliveras, A.: MiniMaxSAT: An efficient Weighted Max-SAT Solver. Journal of Artificial Intelligence Research 31, 1–32 (2008)
8. Jukic, N., Nestorov, S., Vrbsky, S.V., Parrish, A.S.: Enhancing database access control by facilitating non-key related cover stories. Journal of Database Management 16(3), 1–20 (2005)
9. Larrosa, J., Heras, F., de Givry, S.: A logical approach to efficient Max-SAT solving. Artificial Intelligence 172(2-3), 204–233 (2008)
10. Sutcliffe, G.: TPTP, TSTP, CASC, etc. In: Diekert, V., Volkov, M.V., Voronkov, A. (eds.) CSR 2007. LNCS, vol. 4649, pp. 6–22. Springer, Heidelberg (2007)
11. Toland, T.S., Farkas, C., Eastman, C.M.: Dynamic disclosure monitor (D^2Mon): An improved query processing solution. In: Jonker, W., Petković, M. (eds.) SDM 2005. LNCS, vol. 3674, pp. 124–142. Springer, Heidelberg (2005)

A Quantitative Analysis of Indistinguishability for a Continuous Domain Biometric Cryptosystem

Ileana Buhan, Jeroen Breebaart, Jorge Guajardo,
Koen de Groot, Emile Kelkboom, and Ton Akkermans

Philips Research Laboratories, Eindhoven, The Netherlands

Abstract. Biometric information is regarded as highly sensitive information and therefore encryption techniques for biometric information are needed to address security and privacy requirements of biometric information. Most security analyses for these encryption techniques focus on the scenario of one user enrolled in a single biometric system. In practice, biometric systems are deployed at different places and the scenario of one user enrolled in many biometric systems is closer to reality. In this scenario, cross-matching (tracking users enrolled in multiple databases) becomes an important privacy threat. To prevent such cross-matching, various methods to create renewable and indistinguishable biometric references have been published. In this paper, we investigate the indistinguishability or the protection against cross-matching of a continuous-domain biometric cryptosystem, the QIM. In particular our contributions are as follows. Firstly, we present a technique, which allows an adversary to decide whether two protected biometric reference data come from the same person or not. Secondly, we quantify the probability of success of an adversary who plays the indistinguishability game and thirdly, we compare the probability of success of an adversary to the authentication performance of the biometric system for the MCYT fingerprint database. The results indicate that although biometric cryptosystems represent a step in the direction of privacy enhancement, we are not there yet.

1 Introduction

When Alice wants to prove her identity to a biometric authentication system she provides a biometric trait and the system compares the measured biometrics to her reference biometric information. If the two match, Alice is authenticated. For the purpose of authentication, in our model Alice does not need an additional password or token and her reference biometric identity is stored by the authenticating entity. The authenticating entity, however, has to safeguard the privacy of Alice. This important responsibility can be addressed using a variety of requirements and techniques for storing and processing biometric data, for details which include template protection techniques, encryption, etc. we refer to [11]. One of the privacy threats spurred by the widespread use of biometric applications is the ability to track users across applications by comparing biometric references facilitated by the uniqueness and persistance of biometric characteristics. Several counter measures have been identified to prevent cross-matching, which include: (1) the avoidance of central databases by the application of the data separation principle, which recommends storing biometric references on an individual

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 78–92, 2010.

secure token or smartcard, (2) the provision of confidentiality of biometric references by encryption techniques such as DES or AES, and (3) the application of renewable and unlinkable biometric references by means of a diversification process. Renewable and unlikable biometric references correspond to techniques such as discrete fuzzy extractors [4] and continuous fuzzy extractors [1]. Continuous fuzzy extractors are also referred to as biometric cryptosystem [6], while the term biometric template protection often refers to the combination of *all* the *previous* mentioned countermeasures to provide confidentiality, renewability and authenticity for biometric references [11].

In this paper, we investigate the privacy enhancement introduced by a biometric cryptosystem assuming that an attacker has access to protected biometric references in at least two databases. The biometric references are assumed to be protected only by a renewable and preferably unlinkable diversification transform, and additional methods such as data separation or data confidentiality are not used. In our model, the biometric references are protected against abuse in two ways. Firstly, a protected biometric template reveals almost nothing about the biometric characteristics of its owner and, if a database with protected biometric templates is compromised, the attacker cannot learn much about the compromised data. Secondly, if such an intrusion is detected the protected biometric references can be revoked and renewed, since at any time the protection scheme can be reapplied on the original or newly acquired data.

There are two classes of biometric cryptosystems techniques, which are fundamentally different. The first class considers biometric information as discrete variables (a collection of points) and has been formalized by Dodis *et al.* [4] in their definitions for fuzzy extractors and fuzzy sketches. The second class, considers biometric information as continuous variables (probability distributions, which describes the behavior of a user's biometrics over time) and has been formalized by Linnartz and Tuyls [9] and Buhan *et al.* [1]. Both methods use a random, binary string to protect the biometric information. The result of this process is known as sketch.

In this paper, we investigate and quantify the indistinguishability offered by a *continuous* biometric cryptosystems scheme. The scenario is the following: the attacker, Charlie, learns that a particular *protected biometric reference* belongs to Alice. This step is not particularly challenging for Charlie since it is assumed that protected biometric references are public. Now, Charlie would like to know what other accounts Alice has and what information is associated with these accounts. Therefore, the question we ask is: *what is the probability that given a* ***protected*** *biometric reference that belongs to Alice, Charlie can find another protected biometric reference of Alice in a target database?*

OUR CONTRIBUTIONS. Our contributions are threefold. Firstly, we present a technique, which allows an adversary to match a *protected* biometric references generated using a *continuous* method, e.g., Quantization Index Modulation, (QIM) proposed by Linnartz and Tuyls [9] and extended by Buhan *et al.* [2]. Secondly, we quantify indistinguishability by means of the indistinguishability game proposed by Simoens *et al.* [12], and the limitations of this approach are outlined. Thirdly, an alternative, practical evaluation to quantify indistinguishability is described and results for real-world biometric data are provided based on the MCYT fingerprint database.

2 Preliminaries

NOTATION. By capital letters we denote random variables while small letters are used to denote observations of random variables. A random variable is completely described by its probability density function. A random variable A is endowed with a probability distribution $f_A(a)$. With A^d we denote the random variable endowed with a discrete probability distribution $f_{A^d}(a)$ while A^c is used to denote the random variable endowed with the continuous probability distribution $f_{A^c}(a)$. We use the random variable X when referring to a biometric identifier, which is represented as an m-dimensional feature vector. We assume that elements of the feature vector are independent and identically distributed, as commonly assumed in the biometric literature, more details on transformation techniques for biometric data can be found in Duda, *et. al* [5]. Subscripts are used for referring to components of a vector, while superscripts are use for enumerating elements of the same type. The description of the QIM-fuzzy embedder is given for one generic feature element i, which in fact completely describes the whole process due to the independence assumption. The universe of all users with a given biometric identifier is denoted by $\mathcal{U}$. We use variable P when referring to public, protected biometric data, also referred to as a sketch. We use K to denote the key used to protect the biometric data. When referring to noise we use the variable N. We write $[x] = \begin{cases} \lceil x \rceil, \{x\} \geqslant \frac{1}{2} \\ \lfloor x \rfloor, \{x\} < \frac{1}{2} \end{cases}$ for every real number $x \in \mathbb{R}$, whereby $\{x\}$ we denote the fractional part of number x.

QUANTIZATION. A continuous random variable A can be transformed into a discrete random variable by means of quantization, which we write $Q(A)$. Formally, a quantizer is a function $Q : \mathcal{U} \rightarrow M$ that maps each $a \in \mathcal{U}$ into the closest *reconstruction point* in the set $M = \{c_1, c_2, \cdots\}$ using d, an appropriate distance measure defined on $\mathcal{U}$ by

$$Q(a) = \arg\min_{c_i \in M} d(a, c_i), \tag{1}$$

The *Voronoi region* or the *decision region* of a reconstruction point c_i is the subset of all points in $\mathcal{U}$, which are closer to that particular reconstruction point than to any other reconstruction point, with respect to a specific distance measure. We denote with V_{c_i} the Voronoi region of the reconstruction point c_i. When A is one dimensional, Q is called a *scalar* quantizer. If all Voronoi regions of a quantizer are equal in both size and shape the quantizer is *uniform*. In the scalar case, the length of the Voronoi region is then called the *step size*. If the reconstruction points form a lattice, the Voronoi regions of all reconstruction points are congruent. By quantization the probability density function of the continuous random variable A, $f_A(a)$ (which is continuous) is transformed into the probability density function $f_{Q(A)}(a)$ (which is discrete).

HIDING CODES FOR CONTINUOUS VARIABLES. Quantization based data hiding codes as introduced by Chen *et al.* [3] (also known as quantization index modulation) can embed secret information into a real-valued quantity. A *Quantization Index Modulation*, $\mathrm{QIM} : \mathcal{U} \times K \rightarrow M$ data hiding scheme can be seen as a set of individual quantizers $\{Q_1, Q_2, \ldots Q_{2^m}\}$, where $2^m = |K|$ and each quantizer maps $x \in \mathcal{U}$ into a reconstruction point. The quantizer is chosen by the input value $k \in K$. We write $Q_k(x)$ to denote the quantization operation $\mathrm{QIM}(x, k)$. The set of all reconstruction points is

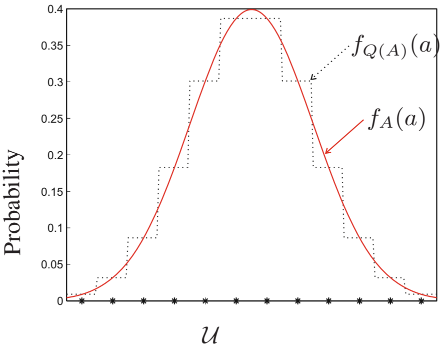


Fig. 1. By quantization, $f_A(a)$ (continuous line) is transformed into $f_{Q(A)}(a)$ (dotted line). We can write $Q(f_A(a)) = f_{Q(A)}(a)$.

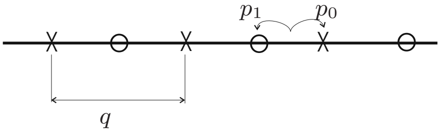


Fig. 2. Quantization of X with two scalar quantizers Q_0 (the set of X points) and Q_1 (the set of o points), corresponding to key bits k_0 and k_1 respectively both with step size q, gives the result, p_0 and p_1 respectively

$M = \bigcup_{k \in K} M_k$ where $M_k \subset M$ is the set of reconstruction points of the quantizer Q_k, and k is known as the *label* of the reconstruction point $Q_k(x)$. The amount of tolerated noise or the reliability is determined by the minimum distance between two neighboring reconstruction points. The size and shape (for high dimensional quantization) of the Voronoi region determines the error tolerance. For the scalar quantizer in the previous example $Q_k(x) = Q_k(y) \text{when} \, d(Q_k(x), y) \leq \frac{q}{2}$. The number of quantizers in the QIM set determines the amount of information that can be embedded. By setting the number of quantizers and by choosing the shape and size of the decision region the performance properties can be finely tuned, more details can be found in Buhan, *et al.* [2].

3 QIM Biometric Cryptosystem

The main challenge in protecting biometric references using cryptographic techniques is coping with noise, which is always introduced into biometric samples during data acquisition and processing. Biometric cryptosystems can transform a noisy, biometric measurement represented as a sequence of non-uniformly distributed real numbers into a reproducible, uniformly-distributed binary string. There are many parameters that control this transformation, for example the length of the output binary sequence, the probability that two measurements coming from the same users will be mapped to the same binary sequence, etc.

Two abstractions, secure sketches and fuzzy extractors were proposed by Dodis, *et al.* [4] to describe the process of transforming a biometric characteristic into a reproducible, uniform binary sequence. A secure sketch can correct the noise between two biometric measurements coming from the same user by using some public information called a sketch. The result of a secure sketch is a reproducible sequence, which is not, necessarily, uniformly distributed and thus not suitable to be used as cryptographic keys. Fuzzy extractors can be used to extract randomness from biometric data to make the output of a secure sketch suitable for usage as cryptographic keys. Both constructions work only on biometric data represented as discrete variables. The process of transforming a continuous variable into a discrete variable influences the performance of fuzzy extractors and secure sketches.

Fuzzy embedders were proposed by Buhan, *et al.* [2] as an extension to the fuzzy extractor idea. A fuzzy embedder can transform a noisy, non-uniform continuous variable, into a reproducible, uniformly random string, which is suitable to be used as a cryptographic key. Basically, the function of a fuzzy embedder is the same as the function of the fuzzy extractor, but its scope is extended so as to accept continuous variables as input. A fuzzy embeder is a pair of procedures. The first is the embed procedure, which is used once when the biometric system learns the identity of the user. The second is the reproduce procedure, which is used to authenticate the user to the server.

QIM BIOMETRIC CRYPTOSYSTEM. Linnartz *et al.* [9] were the first to suggest how to use QIM for the protection of biometric data. The main advantage is that quantization (or discretization) of the biometric data is not required, since QIM works on continuously represented data. Li, *et. al* [8] argue that performance measures like min-entropy or entropy-loss are the result of the quantization parameters used. The larger the quantization step, the less entropy is left in the discrete biometric data and the easier it is to reconstruct the secret k vice-versa the smaller the quantization step, the more entropy remains in the discrete biometric and the harder it is to reconstruct k.

Definition 1 (QIM**-fuzzy embedder [2]).** *A* $(\mathcal{U}, X, K, \eta, m, q)$ - QIM-*fuzzy embedder is a pair of randomized procedures* $<$ Embed, Reproduce $>$ *where*

- Embed *is a function used during enrollment that outputs a sketch* $p \in [-\frac{q}{2}, \frac{q}{2}]^m$ *on input* $k \in K$ *and* $x \in X$*;*
- Reproduce *is a function used during verification that given a word* x' *and any sketch* $p =$ Embed(x, k) *outputs* k *as long as* $d(x_i, x_i') \leq \frac{q}{2}, (\forall) i \in \{1, m\}$.

For any random variable X *over* $\mathcal{U}$ *the probability that an adversary who observes* P *guesses* X *is at most* $\eta = I(X; P)$

For QIM the *enrollment* phase consists of a three step procedure that is applied on each feature vector component x_i separately as shown in *Table* 1.

During *authentication*, a noisy biometric feature vector $x' = (x_1', x_2', \cdots x_m')$ is collected. Verification of a user is performed by reproducing each bit of the biometric key, k_i from the biometric measurement x_i' and the corresponding sketch p_i. The reproduction procedure finds the closest reconstruction point for $Q(x_i' + p_i) \in M$ and returns the label, 0 or 1, associated with this point. The decision to accept or reject a user is done by comparing the obtained key, k' to the enrollment key, k.

Example 1. We want to hide one bit of information, $k \in \{0, 1\}$, into the real value x_i. For this purpose we use a scalar uniform quantizer with step size q, given by rounding x_i to the closest reconstruction point. The public sketch is computed as:

$$Q(x_i) = q\left[\frac{x_i}{q}\right].$$

Table 1. Enrollment and verification algorithm for the QIM, biometric cryptosystem. We observe that the biometric keys k_i and k'_i will be exactly the same as long as $d(x_i, x'_i) \leq \frac{q}{2}$.

Enrollment:	
1.	Generate: $k_i \in \{0, 1\}$;
2.	Apply: $\mathsf{Embed}(x_i, k_i) = Q_{k_i}(x_i) - x_i = p_i$;
3.	Publish: p_i;
Verification:	
1.	$\mathsf{Reproduce}(x'_i, p_i) = k'_i$, where k'_i is the label of the reconstruction point $\left[\frac{x'_i+p_i}{q}\right]q$
2.	If $k_i = k'_i$ accept, otherwise reject;

The quantizer Q is used to generate a set of two new quantizers $\{Q_0, Q_1\}$ defined as:

$$Q_0(x_i) = n(x_i)q \qquad \text{and} \qquad Q_1(x_i) = (n(x_i) + \frac{1}{2})q.$$

In *Figure* 2 the reconstruction points for the quantizer Q_1 are shown as circles and the reconstruction points for the quantizer Q_0 are shown as crosses. The embedding is done by mapping the point x_i to one of the reconstruction points of these two quantizers. For example, if $k = 1$, x_i is mapped to the closest $\circ$ point. Therefore,

$$p_0 = \mathsf{Embed}(x_i, k = 0) = Q_0(x_i) - x_i \;\text{ and }\; p_1 = \mathsf{Embed}(x_i, k = 1) = Q_1(x_i) - x_i$$

where $n(x_i) \in \mathbb{Z}$ is chosen such that $|Q_k(x_i) - x_i| \leq \frac{q}{2}$. The result of the embedding is the distance vector to the nearest $\times$ or $\circ$ as chosen by k. During the reproduction procedure x_i is perturbed by noise then quantizer will assign the received data to the closest $\times$ or $\circ$ point, and output 0 or 1 respectively. The set of the two quantizers $\{Q_0, Q_1\}$ is called a QIM.

Definition 2 (Related Sketches). *Let* $(\mathcal{U}, X, K, \eta, m, q)$ *be a* QIM-*fuzzy embedder. We say that* $p_x = \mathsf{Embed}(x, k)$ *and* $p'_x = \mathsf{Embed}(x', k')$ *are related sketches as long as* $d(x_i, x'_i) \leq \frac{q}{2}$, $(\forall) i \in \{1, m\}$ *for any pair* $\{k, k'\} \in K$.

4 A Theoretical Measure of Indistingishability for the QIM Fuzzy Embedder

n-INDISTINGUISHABILITY. The aim of a biometric cryptosystem, which features the n-indistinguishability attribute as defined by Simoens *et al.* [12] is that no adversary has a significant advantage over random guessing in determining whether n sketches $\{P_1, P_2, \cdots P_n\}$ are related or not. Simoens *et al.* [12] model n-indistinguishability as a game where it is assumed that an adversary has obtained a database of protected biometric references and wants to find the sketches that are related to the reference he holds. As it is the customary in cryptography, the adversary is assumed to know all algorithms used to protect the biometric references. For completeness, we give the description of the game, for two sketches ($n = 2$) below.

1. The challenger randomly selects the variable $X \in \mathcal{U}$ and samples X to obtain $x \in X$. He also selects a secret key $k^{(1)} \in K$ and gives the output of the embed procedure, the sketch P, to the adversary.
2. The challenger flips a fair coin $\mathbf{c} \in \{0, 1\}$. If $\mathbf{c} = 1$, the challenger samples variable X again to obtain x'. If $\mathbf{c} = 0$, the challenger selects another random variable $Y \in \mathcal{U}$ and samples Y to obtain $y \in Y$. Regardless of the result of the coin flip, the challenger selects a new secret key $k^{(2)}$ and gives the output of the embed procedure P', to the adversary.
3. The adversary's aim is to guess correctly whether P' comes from x or y. In particular, the adversary outputs a single bit $\hat{\mathbf{c}} \in \{0, 1\}$ and he wins the game if $\hat{\mathbf{c}} = \mathbf{c}$.

The advantage of the adversary in the indistinguishability game is defined as:

$$Adv_{2-\mathsf{IND}} = 2\left|\mathsf{Pr}[\hat{\mathbf{c}} = \mathbf{c}] - \frac{1}{2}\right| = 2\left|\mathsf{Pr}[\hat{\mathbf{c}} \neq \mathbf{c}] - \frac{1}{2}\right|$$

Notice that we model biometrics as an m-dimensional variable and therefore an adversary who guesses $\hat{\mathbf{c}} = \mathbf{c}$ has to make m correct guesses: $(\hat{\mathbf{c}}_1 = \mathbf{c}_1) \wedge (\hat{\mathbf{c}}_2 = \mathbf{c}_2) \wedge \cdots \wedge (\hat{\mathbf{c}}_m = \mathbf{c}_m)$ one for every component of the public sketch. As we made the assumption that components in the features vector are independent we can write: $\mathsf{Pr}[\hat{\mathbf{c}} = \mathbf{c}] = \prod_{i=1}^{m} \mathsf{Pr}[\hat{\mathbf{c}}_i = \mathbf{c}_i]$ Without loss of generality, in the rest of this section we concentrate on evaluating the advantage of the adversary in the indistinguishability game for one correct guess of the form $\hat{\mathbf{c}}_i = \mathbf{c}_i$, and all definitions are given for a $(\mathcal{U}, X, K, \eta, 1, q)$ QIM-fuzzy embedder. The adversary in the above game is called $\mathsf{Charlie}_{\mathsf{IND}}$ and his advantage in the game is defined as:

$$Adv^{i}_{2-\mathsf{IND}} = 2\left|\mathsf{Pr}[\hat{\mathbf{c}}_i = \mathbf{c}_i] - \frac{1}{2}\right| = 2\left|\mathsf{Pr}[\hat{\mathbf{c}}_i \neq \mathbf{c}_i] - \frac{1}{2}\right| \qquad (2)$$

Definition 3 (ϵ-Indistiguishability). *An* $(\mathcal{U}, X, K, \eta, 1, q)$ QIM-*fuzzy embedder* $< \mathsf{Embed}, \mathsf{Reproduce} >$ *is ϵ-indistinguishable if for any adversary* $\mathsf{Charlie}_{\mathsf{IND}}$*, such that $Adv_{2-\mathsf{IND}} = Adv_{\mathsf{Charlie}_{\mathsf{IND}}}$ it holds that $Adv^{i}_{2-\mathsf{IND}} \leq \epsilon$.*

Definition 4 (QIM-Distinguisher). *For any two sketches p_{x_i} and p_{y_i} generated by an $(\mathcal{U}, X, K, \eta, 1, q)$* QIM-*fuzzy embedder* $< \mathsf{Embed}, \mathsf{Reproduce} >$ *the function $\mathcal{H}^{\delta}$, defined as:*

$$\mathcal{H}^{\delta}(p_{x_i}, p_{y_i}) = \begin{cases} 1, \text{ if } |p_{x_i} - p_{y_i}| \leq \delta, \text{ or } |p_{x_i} - p_{y_i} - \frac{q}{2}| \leq \delta, \text{ or } |p_{x_i} - p_{y_i} + \frac{q}{2}| \leq \delta; \\ 0, \text{ otherwise.} \end{cases}$$

is a QIM-*distinguisher.*

A few explanations are in order to motivate the introduction of the parameter δ in the definition of the distinguisher. For an average user Alice, the distance between two random samplings x and x' of variable X is at most $\frac{q}{2}$, $d(x_{\mathsf{Alice}}, x'_{\mathsf{Alice}}) \leq \frac{q}{2}$. However, if Charlie, knows that the biometric data of the user he is targeting for cross-matching (Dave) is better (less noise between different samplings) compared to that of the average user (Alice), Charlie has an additional advantage. We model this advantage by the introduction of the parameter δ. By choosing a value δ Charlie has control over the distance between two biometric measurements of Dave, $d(x_{\mathsf{Dave}}, x'_{\mathsf{Dave}}) \leq \delta \ll \frac{q}{2}$.

Lemma 1 (Distinguishing related sketches.). *Let x_i and x'_i be two samples of random variable X, furthermore let $x'_i = x_i + \delta_i$, with $|\delta_i| \leq \frac{q}{2}$. For any, two related sketches p_{x_i} and $p_{x'_i}$ generated by an $(\mathcal{U}, X, K, \eta, 1, q)$ QIM-fuzzy embedder $< \mathsf{Embed}, \mathsf{Reproduce} >$ the QIM-distinguisher always outputs the value 1.*

Proof. To make the proof more readable, we firstly analyze the simple case when the sampling of variable X, yields $x_i = x'_i$. This case corresponds to the scenario when there is no noise between different enrollment samples of user X. Secondly we extend the simple case to the scenario when different enrollment samples, x and x' of user X are subjected to noise, $d(x_i, x'_i) \leq \delta$. For both cases we derive the value of the difference $p_{x_i} - p_{y_i}$ when the two sketches are related and we show that $\mathcal{H}^{\delta=0}(p_{x_i}, p_{y_i})$ and $\mathcal{H}^{\delta}(p_{x_i}, p_{y_i})$ are equal to 1 in both cases.

Simple case: ($x_i = x'_i$). Although the different keys $k^{(1)}$ and $k^{(2)}$ are used to generate sketches for x and x' respectively, we discovered there is a simple test to verify whether the resulting sketches $p^{(x)} = (p_{x_1}, p_{x_2}, \cdots p_{x_m})$ and $p^{(x')} = (p_{x'_1}, p_{x'_2}, \cdots p_{x'_m})$ are related. Each element p_{x_i} and $p_{x'_i}$ of the public sketches is computed as: $p_{x_i} = \mathsf{Embed}(x_i, k_i^{(1)}) = Q_{k_i^{(1)}}(x_i) - x_i$ and $p_{x'_i} = \mathsf{Embed}(x'_i, k_i^{(2)}) = Q_{k_i^{(2)}}(x'_i) - x'_i$, where quantization is defined in equation (1). In deriving a distinguishing function, the adversary can distinguish three cases:

Case I: The result of the coin flip is $\mathbf{c}_i = 1$, $(x_i = x'_i)$ and the key bits are equal $(k_i^{(1)} = k_i^{(2)})$. By subtracting the two sketches the adversary obtains:

$$p_{x_i} - p_{x'_i} = (Q_{k_i^{(1)}}(x_i) - x_i) - (Q_{k_i^{(1)}}(x_i) - x_i) = 0;$$

Case II: The result of the coin flip is $\mathbf{c}_i = 1$, $(x_i = x'_i)$ however the key bits are different $(k_i^{(1)} \neq k_i^{(2)})$. By subtracting the two sketches the adversary obtains:

$$\begin{aligned} p_{x_i} - p_{x'_i} &= Q_{k_i^{(1)}}(x_i) - x_i - (Q_{k_i^{(2)}}(x_i) - x_i) \\ &= Q_{k_i^{(1)}}(x_i) - Q_{k_i^{(2)}}(x_i) = \pm\frac{q}{2} \end{aligned}$$

Figure 2 shows that embedding two different bits in the same value will lead always to sketches that are complementary, $p_0 - p_1 = \pm\frac{q}{2}$.
Case III: The result of the coin flip is $\mathbf{c}_i = 0$ and $x_i \neq x'_i$ when subtracting two sketches the result is different from 0 or $\pm\frac{q}{2}$.

To summarize, by subtracting p_{x_i} and p_{y_i} we obtain

$$p_{x_i} - p_{y_i} = \begin{cases} 0, & \text{if } (\mathsf{c}_i = 1) \wedge (k_i^{(1)} = k_i^{(2)}) \\ \pm\frac{q}{2}, & \text{if } (\mathsf{c}_i = 1) \wedge (k_i^{(1)} \neq k_i^{(2)}) \\ p_{x_i} - p_{y_i} & \text{if } (\mathsf{c}_i = 0); \end{cases}$$

Therefore when p_{x_i} and p_{y_i} are related $|p_{x_i} - p_{y_i}| \in \{0, \frac{q}{2}\}$ and $\mathcal{H}^{\delta=0}(p_{x_i}, p_{y_i}) = 1$.

General Case ($d(x_i, x'_i) \leq \delta$). During enrollment, multiple measurements for the same individual are taken. The average of these measurements is computed and stored as reference information. Due to the unpredictable amount of noise existent in each measurement the reference information changes as well. The extended case, models this scenario by assuming that the biometric reference information of person X gives two different reference values x_i and x'_i that are within distance δ and therefore we set $d(x_i, x'_i) \leq \delta$. Derivation of function $|p_{x_i} - p_{y_i}|$ is straightforward, by replacing $x' = x + \delta$ in the three cases derived in the previous paragraph. As a result we obtain:

$$|p_{x_i} - p_{y_i}| \in \begin{cases} (-\delta, \delta), & \mathsf{if}(\mathbf{c}_i = 1) \wedge (k_i^{(1)} = k_i^{(2)}) \\ (-\frac{q}{2} - \delta, -\frac{q}{2} + \delta) \cup (\frac{q}{2} - \delta, \frac{q}{2} + \delta) & \mathsf{if}(\mathbf{c}_i = 1) \wedge (k_i^{(1)} \neq k_i^{(2)}) \\ p_{x_i} - p_{y_i} & \mathsf{if}(\mathbf{c}_i = 0); \end{cases} \quad (3)$$

Therefore when p_{x_i} and p_{y_i} are related $\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 1$. □

In fact, x_i and x'_i can be samples from different distributions, as long as the conditions of lemma 1 are satisfied, the distinguisher returns 1.

Lemma 2 (ϵ-Indistinguishability). *An* $(\mathcal{U}, X, K, \eta, 1, q)$ QIM-*fuzzy embedder* $<\mathsf{Embed}, \mathsf{Reproduce}>$ *is ϵ-indistinguishable for any adversary* $\mathsf{Charlie}_{\mathsf{IND}}$*, and it holds that:* $\left| \int_\Delta f_{D_{P_i}}(p(t))dt - 1 \right| \geq \epsilon$ *where* $\Delta = (-\frac{q}{2} - \delta, -\frac{q}{2} + \delta) \cup (-\delta, \delta) \cup (\frac{q}{2} - \delta, \frac{q}{2} + \delta)$ *and* $f_{D_{P_i}}$ *is the probability distribution of the difference between* $P_{x_i} - P_{y_i}$ *and* P_{x_i}, P_{y_i} *are the random variables from which* p_{x_i} *and* p_{y_i} *are sampled.*

Proof. If the adversary uses $\mathcal{H}^\delta$ for guessing the challengers coin flip he will always guess that p_{x_i}, p_{y_i} are not related when $\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 0$, regardless of the coin flip. The adversary also guesses that p_{x_i}, p_{y_i} are related when $\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 1$ and $\mathbf{c}_i = 1$. The adversary makes an incorrect guess when the coin flip is $\mathbf{c}_i = 0$, (the sketches are not related) but $\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 1$. It follows that the probability of an incorrect guess can be derived from: $\mathsf{Pr}[\hat{\mathbf{c}}_i \neq \mathbf{c}_i] = \mathsf{Pr}[\hat{\mathbf{c}}_i = 0|\mathbf{c}_i = 1]\mathsf{Pr}[\mathbf{c}_i = 1] + \mathsf{Pr}[\hat{\mathbf{c}}_i = 1|\mathbf{c}_i = 0]\mathsf{Pr}[\mathbf{c}_i = 0]$. According to lemma 1, the distinguisher always returns 1 when the sketches are related (FRR=0). Therefore, the probability that the adversary guesses $\hat{\mathbf{c}}_i = 0$, when $\mathbf{c}_i = 1$, is 0. Therefore $\mathsf{Pr}[\hat{\mathbf{c}}_i = 0|\mathbf{c}_i = 1] = 0$. The probability that the adversary guesses $\hat{\mathbf{c}}_i = 1$ when $\mathbf{c}_i = 0$ is: $\mathsf{Pr}[\hat{\mathbf{c}}_i = 1|\mathbf{c}_i = 0] = \mathsf{Pr}\left[\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 1 \middle| \mathbf{c}_i = 0\right]$ When knowing the probability distribution of the sketch P_i, denoted by $f_{P_i}(p)$, we can compute probability distribution for the difference between variables $D_{P_i} = P_{x_i} - P_{y_i}$, we can write:$\mathsf{Pr}[\mathcal{H}^\delta(p_{x_i}, p_{y_i}) = 1|\mathbf{c}_i = 0] = \int_\Delta f_{D_{P_i}}(p(t))dt$. The probability that the adversary makes an incorrect guess is therefore:

$$\mathsf{Pr}[\hat{\mathbf{c}}_i \neq \mathbf{c}_i] = \frac{1}{2} \int\limits_\Delta f_{D_{P_i}}(p(t))dt. \quad (4)$$

Substitutions equation (4) in equation (2), proves $\epsilon = \left| \int_\Delta f_{D_{P_i}}(p(t))dt - 1 \right|$. An adversary with an improved strategy or a superior distinguisher has an advantage that is larger compared to ϵ which completes our proof. □

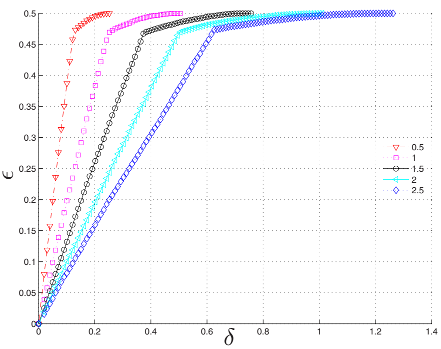


Fig. 3. $\mathsf{Pr}[\hat{\mathbf{c}}_i \neq \mathbf{c}_i]$ for different q and $\mathcal{U} = N(0, 1.)$

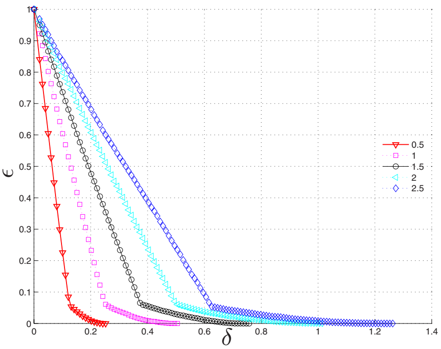


Fig. 4. Advantage of an adversary for ant q and $\mathcal{U} = N(0, 1)$

EVALUATION OF THE ADVERSARY ADVANTAGE. Figure 4 shows the bounds on ϵ-Indistinguishability for an $(\mathcal{U}, X, \eta, 1, q)$ fuzzy embedder for several quantization steps q, when the probability distribution of $\mathcal{U}$ is modeled as a normal distribution with $\mu = 0$ and $\sigma = 1$. We conclude that the advantage of Charlie in the 2-Indistinguishability game is significant and it increases with the quantization step (which is preferred in practice as it increases the classification performance of the biometric authentication). Another interesting observation is that, Charlie has a larger advantage when targeting a person which has very stable biometric identifiers. This means that an adversary can easily identify protected biometric references which come form the same person and the more stable the biometric the more accurate is his identification (Dave vs. Alice).

In the next section we use the distinguisher $\mathcal{H}^\delta$, as defined in this section to execute a cross-matching attack on real biometric data.

5 A Practical Measure of Indistinguishability for the QIM Fuzzy Embedder

Although the concept of n-indistinguishability is very suitable to describe the theoretical advantage of an attacker, it has its limitations. The concept describes the advantage of an attacker with respect to a perfectly indistinguishable system. On the one hand, perfect indistinguishability is hard to achieve when employing only biometric cryptosystems due to the inherit correlation of the data used to generate related biometric references. On the other hard, indistinguishability is *not* hard to achieve when biometric template protection techniques are employed. The protected biometric reference is encrypted and the encryption key is stored outside the database, for example on a smartcard. A user who wishes to verify his identity uses the key stored on the smartcard to decrypt the sketch and then proceeds to perform biometric authentication. The ciphertext indistinguishability of most encryption schemes guarantees that the biometric sketches achieve indistinguishability. Therefore, in this section we investigate the other side of the indistinguishability game presented in the previous section, namely *is there a gain in privacy with respect to cross-matching, when using a biometric cryptosystem?*

To answer this question we compare the classification performance of the MCYT fingerprint database in two distinct scenarios. *Scenario I* models the classification performance of the QIM-fuzzy embedder during normal operation. User x is first enrolled in the biometric system and the public sketch p_x is computed as $p_x = \mathsf{Encode}(x, k)$. During verification the user presents his biometric x' and the server computes $\mathsf{Reproduce}(x', p_x) = k'$. Authentication is considered successful if the Hamming distance between k and k' is zero. For a QIM-fuzzy embeder the percentage of successful authentication depends on the distance tolerated between x and x', which is a function of the quantization step,q, given by $d(x, x') \leq \frac{q}{2}$. *Scenario II* corresponds to the scenario when the adversary has access to the protected biometric references, $p_x = \mathsf{Embed}(\mathsf{x}, \mathsf{k_x})$ and $p_y = \mathsf{Embed}(\mathsf{y}, \mathsf{k_y})$. In this case, classification performance is evaluated using the distinguisher function, $\mathcal{H}^{\delta}(p_x, p_y)$ constructed in Section 4.

We propose to use cross-matching performance differences between unprotected and protected biometric references as relevant measure for indistinguishability. These properties can be described by a receiver operating characteristic curve (ROC) indicating false match and false non-match rates. In this section, for the evaluation we use the MCYT database [10], which is known in the literature as a good quality data set and has good classification performance. In the current context of testing for indistinguishability a good quality database is rather a pessimistic choice. We expect that the sketch classification performance improves with good quality data (less noise expected between biometric references collected from the same person).

The MCYT database consists of fingerprints collected from 323 individuals. For each individual, 12 fingerprints images have been captured under the supervision of an operator. Fingerprint images were collected with an optical sensor (Digital Persona), which gives as output images having resolution of 256×400 pixels and 8 bit gray-scale levels. From the total of 323 individuals, 80% are used for training the algorithm and the rest of 20% (approximately 66 individuals) are used for testing the performance of the algorithm. During testing, the data is split into two sets. The first set consisting of 4 fingerprints are used as enrollment data. The second set consisting of 8 fingerprints is used as verification set. For each round of experiments 5 random splits are performed on the testing data and the results are averaged.

Each fingerprint in the database is processed and represented as a fixed length vector. To describe the shape of the fingerprint, two types of features are extracted. The first feature vector is the squared directional field and the second feature vector is the Gabor response of the fingerprint, details can be found in Tuyls *et al.* [13]. The resulting feature vector is a concatenation of the squared directional field and the Gabor response and describes the global shape of the fingerprint in 1536 elements. Prior to applying QIM, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) transformations are applied on each continuous-domain feature vector to reduce its dimensionality to the desired length while maintaining maximum discrimination. The PCA and LDA parameters are obtained from the training set. The enrollment feature vector is constructed by averaging the set of enrollment feature vectors.

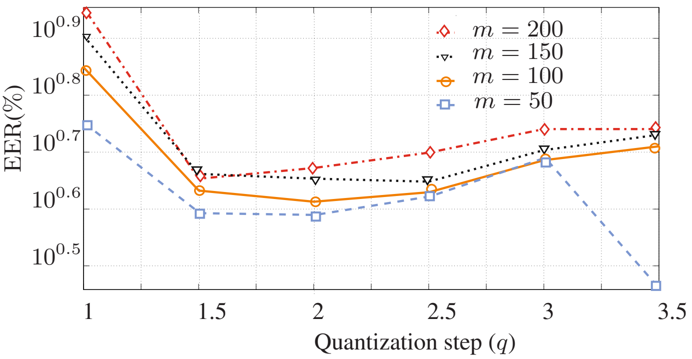


Fig. 5. (Scenario I) The classifier used is the Hamming distance between k and $k' = \mathsf{Reproduce}(x, p_x)$, where $p_x = \mathsf{Embed}(x, k)$. The EER value, expressed in percent, for each of the experiments is plotted, as a function of the amount of noise tolerated between biometric samples of the same individual, $d(x, x') \leq \frac{q}{2}$.

There are two dual measures in the biometric literature to measure resilience to noise. The first is *False Rejection Rate* (FRR), which estimates the probability that the public sketch of person A and a measurement of person A produce a faulty secret key. The second measure is *False Acceptance Rate* (FAR), which estimates the probability that a public sketch of person A and a measurement of person B produce the correct secret key of person A. For QIM the factors that influences the FAR and FRR values (besides the quality of the data, which is determined by the impostor versus genuine standard deviation) are: (a) the quantization step q, which determines the amount of noise tolerated between biometric measurements of the same individual, $d(x, x') \leq \frac{q}{2}$ and (b) the number of features m that are used. Different values for the FAR and FRR are obtained by varying the maximum accepted Hamming distance between measurements coming from the same person. This curve is called *Receiver Operating Characteristic* curve (ROC). The point where the FAR and the FRR are equal is known as the *Equal Error Rate* (EER) and is used as reference point.

The first set of experiments, corresponds to *Scenario I* and measures the performance of the biometric recognition algorithm in a classical use scenario: user x is first enrolled in the biometric system and the public sketch p_x is computed as $p_x = \mathsf{Encode}(x, k)$. During verification the user presents his biometric x' and the server computes $\mathsf{Reproduce}(x', p_x) = k'$. *Figure* 5 shows the EER for different quantization steps (q) and different number of features (m), when the Hamming distance is used to compute the distance between $\mathsf{Reproduce}(x, p_x)$ and $\mathsf{Reproduce}(x', p_x)$. As expected, the smaller the quantization step, the less noise is tolerated and the higher the EER. For example the EER goes up from 5.38% for $q = 3.5$ to 8.03% for $q = 1$ (for $m = 150$ features). Also, the more features, the less accurate the classification performance becomes. For example, for $m = 200$ features the EER is approximately 5.5% while for $m = 50$ the error rates are significantly lower, approximately 2.73% (for q=3.5). The reason for plotting the curves in *Figure* 5 is to have a reference for the classification performance of the public sketches. The second set of experiments, correspond to

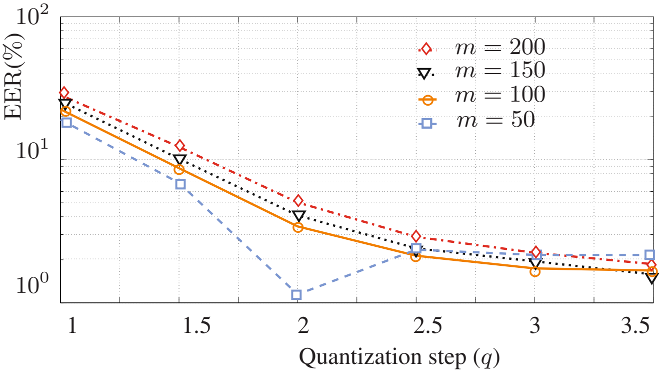


Fig. 6. (Scenario II) The classifier used in this case is the QIM-distinguisher $\mathcal{H}^{\frac{q}{4}}(p_x, p_y)$ given in definition 4. The EER values are obtained by varying the length of the feature vector m, and the size of quantization step q. The EER value, expressed in percentage, for each of the experiments is plotted, as a function of the amount of noise tolerated between biometric samples of the same individual, $d(x, x') \leq \frac{q}{2}$.

Scenario II and measure the performance of the cross-matching algorithm. The classifier used in this case is the distinguishing function $\mathcal{H}^{\delta}(p_x, p_y)$, where p_x is a sketch found in the first database and p_y is a sketch found in the second database. The two databases are obtained by randomly splitting the MCYT database. The result of $\mathcal{H}^{\delta}(p_x, p_y)$, is a binary string of length m (the number of components in p_x and p_y, respectively) where each bit is obtained from equation (3). As shown in *Figure* 6 the EER goes down from 28.4% for $q = 1$ to only 1.86% for $q = 3.5$ ($m = 200$ features). Also, the more features are used, the less accurate the classification performance becomes. For example the 1.15% EER obtained for $m = 50$ features increases to 5.03% EER for $m = 200$ features (for $q = 2$).[1] The results were obtained for setting $\delta = \frac{q}{4}$, see *Section* 4. As expected it seems that the same settings that improve the classification performance of the QIM biometric cryptosystem improve the sketch classification performance. In other words linking users across databases becomes easier when the biometric classification performance improves. Comparing the classification results obtained in *Scenario I* and *Scenario II* we conclude that employing biometric cryptosystems improves, the biometric sketch indistinguishability in most cases. For example, for m=200 features, the classification performance for $q = 1$ is 9% in *Scenario I* decreases to 28.40%, for the same quantization step, and m=100 features, the classification performance is 7.05% in *Scenario I* and 22.56% in *Scenario II*.

The surprising result of these experiments is that, not only that indistinguishability is not achieved for all quantization steps but in some cases the sketch classification performance (*Scenario 2*), see *Figure* 6 offers better performance compared to the biometric classification (*Scenario 1*), see *Figure* 5. We explain this phenomenon by the fact that in the classical biometric classification a 4:1 matching (4 measurement used during enrollment and 1 measurement used for authentication) is employed. On the other hand

[1] We note that a classification performance of 1.15% means that an attacker can guess with 98.85% probability whether two sketches are related or not.

matching sketches represent a 4:4 comparison (4 measurements are used when computing the sketches during enrollment) and thus sketch classification is less corrupted by noise. Kelkboom, *et al.* [7] show that a 4:4 matching (4 enrollment measurements:4 verification measurements) has superior classification results compared to a 4:1 matching (4 enrollment measurements:1 verification measurement). This supports, from a theoretical perspective the results we obtained in practice. We consider these settings to be realistic as current practice in the field is to collect multiple samples during enrollment and less samples during verification. The main conclusion is positive in the sense that biometric cryptosystem have a positive effect on privacy, with respect to cross-matching, however we seem to have a lot to improve in this sense.

6 Conclusions

Privacy compliant databases should ensure that users are indistinguishable. In this paper, we show how an adversary can distinguish between protected biometric references generated with the QIM-fuzzy embedder. In this context we show that the advantage of an adversary who plays the indistinguishability game is non-negligible. Secondly, we look at the indistinguishability property from a practical perspective. We first randomly split the MCYT fingerprint database into two databases such that each user in the MCYT database can be found both databases. We apply the QIM fuzzy embedder for each user using different random sequences to protect the reference biometric samples in the two different databases. On the protected references we apply the distinguishing function, to determine whether they belong to the same user or not. As the performance of the cross-matching attack depends on the amount of noise tolerated between different samplings of a users biometric we compare the results to the error rates of the normal operation point. The results indicated that the QIM method does provide a certain amount of cross-matching resilience, but at the same time does not meet the desired requirement of complete unlinkability (and hence indistinguishability) when using the MCYT fingerprint database.

Acknowledgements

We would like to thank Willem Jonker and Raymond Veldhuis for their support in presenting this work.This work has been funded in part by the European Community's Sixth Framework Programme under grant number 034238, SPEED project - Signal Processing in the Encrypted Domain. This work is partly supported by funding under the Seventh Research Framework Programme of the European Union, Project TURBINE (ICT- 2007-216339). The work reported reflects only the authors views; the European Community is not liable for any use that may be made of the information contained herein.

References

1. Buhan, I.R., Doumen, J., Hartel, P.H., Veldhuis, R.N.J.: Fuzzy extractors for continuous distributions. In: Deng, R., Samarati, P. (eds.) Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS), Singapore, pp. 353–355. ACM, New York (2007)

2. Buhan, I.R., Doumen, J., Hartel, P.H., Veldhuis, R.N.J.: Embedding renewable cryptographic keys into continuous noisy data. In: Chen, L., Ryan, M.D., Wang, G. (eds.) ICICS 2008. LNCS, vol. 5308, pp. 294–310. Springer, Heidelberg (2008)
3. Chen, B., Wornell, G.W.: Quantization index modulation methods for digital watermarking and information embedding of multimedia. The Journal of VLSI Signal Processing 27(1-2), 7–33 (2001)
4. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
6. Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security. Journal on Advances in Signal Processing (EURASIP) 2008, 17 (2008)
7. Kelkboom, E., Garcia Molina, G., Breebaart, J., Kevenaar, T.A.M., Veldhuis, R.N.J., Jonker, W.: Binary biometrics: An analytic framework to estimate the performance curves under gaussian assumptions. IEEE Transactions on Systems, Man and Cybernetics (to appear, 2009)
8. Li, Q., Sutcu, Y., Memon, N.: Secure sketch for biometric templates. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 99–113. Springer, Heidelberg (2006)
9. Linnartz, J.P., Tuyls, P.: New shielding functions to enhance privacy and prevent misuse of biometric templates. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 393–402. Springer, Heidelberg (2003)
10. Ortega-Garcia, J., Fierrez-Aguillar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.-J., Vivaracho, C., Escudero, D., Moro, Q.-I.: Myct baseline corpus: a bimodal biometric database. In: IEEE Proceedings on Vision, Image and Signal Processing, Special Issue on Biometrics on the Internet, vol. 150, pp. 395–401. IEEE Computer Society Press, Los Alamitos (2003)
11. ISO/IEC JTC1 SC27. CD 24745 - information security - biometric template protection
12. Simoens, K., Tuyls, P., Preneel, B.: Privacy weakness in biometric sketches. In: IEEE Symposium on Security and Privacy, Oakland, California, USA (May 2009)
13. Tuyls, P., Akkermans, A., Kevenaar, T., Schrijen, G., Bazen, A., Veldhuis, R.: Practical biometric authentication with template protection. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 436–446. Springer, Heidelberg (2005)

A Spatial Cloaking Framework Based on Range Search for Nearest Neighbor Search

Hyoungshick Kim

Computer Laboratory,
University of Cambridge, UK
hk331@cam.ac.uk

Abstract. For nearest neighbor search, a user queries a server for nearby points of interest (POIs) with his/her location information. Our aim is to protect the user's sensitive information against adversaries including the location-based service itself. Most research efforts have elaborated on reasonable trade-offs between privacy and utility. We propose a framework based on range search query without a trusted middleware. We design a query processing algorithm for the minimum set of candidate POIs by computing the local Voronoi diagram relevant to the cloaked region. Contrary to common belief that cloaking approaches using range search incur expensive processing and communication cost, the experimental results show that the framework incurs reasonable processing and communication overhead even for large cloaked regions.

Keywords: Location Anonymity, Spatial Cloaking, Query Privacy, Voronoi Diagram, Nearest Neighbor Search.

1 Introduction

With the rapid evolution of mobile computing, location sensing, and wireless networking, geospatial applications are quickly growing in popularity [1]. Location-based services are personalized services in geospatial applications to provide useful location information for a given position. One of fundamental location-based services is to search the nearest neighbor to user location. A user can ask the closest POIs (e.g., hospital, hotel, or gas station) to her current location.

For personalized location-based services, a user must report her location. Location is an especially sensitive type of personal information. The information about user location may be clue to infer the user's sensitive information such as health, private lifestyle, and personal preference. For example, an employer may check on her employee's behaviour by knowing the places the employee visits and the time of each visit, the personal medical records can be inferred by knowing which the clinic a person visits, or someone can stalk the locations of her acquaintances. Therefore location privacy will be one of the key issues to deploy location-based services although they provide helpful and intelligent results.

As an intuitive approach to preserving location privacy, we enlarge an exact user location into a cloaked region so that it is infeasible to infer the user's exact

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 93–105, 2010.

location from the cloaked region. Sensitive information about an individual user location can be protected by controlling the level of detail of a cloaked region including user location. Previously, it was believed that spatial cloaking solutions using range search query incur expensive processing and communication cost for large number of POIs. However, we believe that range-based spatial cloaking can be practically applicable since recent growth in networking technology have enabled communication to transmit high bandwidth data (e.g., map data) in real time.

We propose a range-based framework that does not rely on an external anonymizer, which collects the location information of users and anonymizes their queries. In practice, it is hard to assume a trustful mediator between users and location services. Since most existing location-based services are based on a standard client-server architecture, it is desirable for two-tier spatial cloaking where the cloaked region can be constructed and sent by the user directly without dependency of an external party.

In range-based spatial cloaking, the most challenging issue is to minimize processing and communication overheads due to range search. There is an inherent trade-off between user privacy and service utility. A larger cloaked region implies higher guarantees for location privacy, but it also requires high computational and communication costs.

Interestingly, given a cloaked region including user location, finding the nearest POI to the user location cannot be acheived by range search with a fixed region. Fig. 1 illustrates that the problem of range search with a fixed region. In this example, the nearest POI to the user location u is p_1. The conventonal range search algorithms with a fixed region, that do not consider outer points of the region, cannot guarantee the nearest POI to user location.

Therefore, we should also consider outer points of a cloaked region. We observe that it can be transformed to finding intersections of Voronoi cells for POIs with a cloaked region since the user position can be uniformly located at the cloaked region. This also means that the minimum size of the candidate answer results in $\Omega(k)$ where k is the number of the Voronoi cells which intersect with a cloaked region. Our query processing algorithm is based on computation of the intersections of Voronoi cells with a cloaked region.

When the locations of POIs (e.g., buses) are dynamically changed or the server's storage is limited to maintain the overall Voronoi diagram, the precomputed Voronoi diagram cannot be used. Our objective is to avoid computing

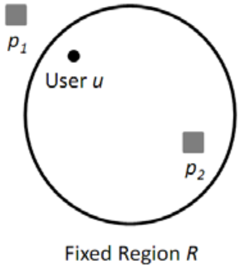


Fig. 1. A counter-example to range search with a fixed region

the global Voronoi diagram for a large data set, which is forbiddingly costly in terms of CPU and memory. Therefore we design the online computation of the candidate neighbors using the local Voronoi diagram relevant to a cloaked region. We show that the computed local Voronoi diagram always successfully include the correct query answer.

In addition, we suggest a heuristic sampling method to provide an approximate answer statistically when a limited communication bandwidth is required for nearest neighbor search. A reasonable approximate sample set can be also retrieved using the intersections of Voronoi cells depending on the maximum permitted communication cost.

The proposed framework is simple and can be integrated into a general server-client architecture without a trusted middleware. Empirical studies show reasonable communication cost in real datasets even if a user requires high privacy.

The remainder of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we introduce data structure, notations, and threat model. In Section 4, we propose a framework based on computation of local Voronoi diagram. The experimental results in terms of communication and computational costs, are analysed in Section 5. Finally, we conclude the results and suggest some directions for future research in Section 6.

2 Related Work

In spatial cloaking, user location is enlarged into a cloaked region that is then used for querying the server. One of the main goals in those studies is to provide k-anonymity. The concept of k-anonymity was originally introduced in the context of relational data privacy [25,23]. The k-anonymity model with respect to location information was defined as follows: A query message from a user to a server is called k-anonymous in location-based services if the user cannot be identified by the server based on the user location from the other $k-1$ users where k is a user-specified anonymity set size [7].

A trusted third party called anonymizer is basically required to achieve k-anonymity with respect to location information since it is hard to construct a cloaked region including k users' queries in a distributed manner [28]. In order to provide k-anonymity, many techniques [12,18,3,10] were proposed based on the assumption of a trusted anonymizer. Fig. 2 illustrates a three-tier architecture with a trusted anonymizer. All queries and answers are relayed through the anonymizer. Given a query, the anonymizer removes the user's identifier, applies cloaking to replace the user location with a cloaked region, and then forwards the cloaked region to the location server.

However, in real applications, the assumption of a trusted anonymizer is not desirable. First of all, we should consider major redesign of technologies (e.g., protocols or trusted mechanism) or business models. It may be not easy to share private service information including map or POIs with other business entities including the anonymizer since the information in location-based services is generally valuable. Second, we should consider the problems inherent in a

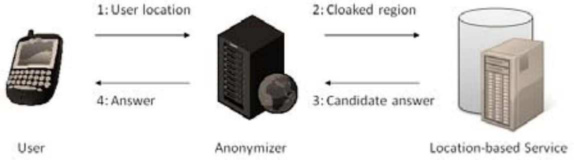


Fig. 2. Three-tier architecture with a trusted third party

single server design since query and response process should always be processed through trusted third party. Moreover, the anonymizer is a single point of attack: if an adversary gains access to it, the privacy of all users is compromised. Third, a large number of users must subscribe to the service, otherwise the cloaked region may not be constructed. It is also assumed that all users are trustworthy. If some of them are compromised, the privacy of a targeted user may be threatened.

In order to overcome the limitation of three-tier architecture, several research efforts are dedicated to constructing a cloaked region at the user (e.g., false dummies [14], landmark objects [11], location perturbation [5,28], transformation-based matching using an obfuscated map [13,16] and transformation-based matching using Private Information Retrieval (PIR) [9]). In the spatial cloaking using dummies, however, the adversary approximately estimates the user location with high accuracy by using cellular positionning techniques [22] or target tracking. In the cloaked region using landmark objects, the accuracy of the answer cannot be generally guaranteed. Transformation-based matching using an obfuscated map also requires a trusted entity that creates an obfuscated map. Transformation-based matching using PIR theoretically provides high privacy but it incurs significant communication and computational overheads compared to other solutions [8].

SpaceTwist [28] incrementally updated the candidate answer without the fixed cloaked region using a faked location called anchor location which is initially set to a location randomly generated by the user. However it consists of multiple message rounds, which may lead to increased response time. Moreover, it cannot always guarantee the user's desired level of privacy.

3 Preliminaries

In this section, we first introduce Voronoi diagram, which is used as a basic data structure in the proposed framework and then define the notations, and threat model.

3.1 Voronoi Diagram

Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of n points (called sites) in the multi-dimensional Euclidean space. We define the Voronoi diagram of P as the subdivision of the space into n cells, one for each site in P, with the property that a point q lies in the cell corresponding a site p_i if and only if the $dist(q, p_i) < dist(q, p_j)$ for each $p_j \in P$ with $j \neq i$ where $dist$ denotes the euclidean distance function.

We denote the Voronoi diagram of P by $Vor(P)$. The cell of $Vor(P)$ that corresponds to a site p_i is denoted $V(p_i)$; we call it the Voronoi cell of p_i [4].

3.2 Notations

The symbols U and L represent a user and a location server, respectively. The symbol q represents a query position and N a set of the nearest neighbors. The subscript X in N_X implies that a POI in N_X is the nearest neighbor from any point within the region X. A is a function to compute a cloaked region with q and s where q is randomly located on the region $A(q, s)$ and s is a security parameter which is relevant to the level of privacy. For example, $A(q, s)$ is a disc with the radius of s. $D(P)$ is a function to compute the smallest enclosing disc for a set of points P.

3.3 Threat Model

In our model, the adversary is attempting to infer user location by monitoring the communication between a user and a service. Each user has its own privacy requirement A_{min} that specifies its desired level of privacy. A_{min} specifies the minimum resolution of the cloaked spatial region. Our goal is to protect the information about user location so that the adversary only knows the region A in which the user is located, but not her exact location in A where the size of A is greater than A_{min}.

4 The Proposed Framework

The proposed framework is basically based on processing of range search query.

4.1 Protocol

The protocol between U and L is briefly described in Algorithm 1. The cloaked region $A(q, s)$ is generally regarded as a convex polygon with m vertices. $A(q, s)$ can be simply computed as a disc with the radius of s where s is the half of the diameter of A_{min}. The proposed protocol results in $O(k \cdot l_1 + m \cdot l_2)$ bits of communication where k is the number of Voronoi cells which intersect with $A(q, s)$ and l_1 and l_2 are the minimum bits to encode a point and a vertex of a polygon, respectively. L can compute $N_{A(q,s)}$ using the Voronoi diagram for POIs. Finally, after receiving the query response $N_{A(q,s)}$, U can find the nearest POI N_q, by filtering out the false positives from $N_{A(q,s)}$ since $N_{A(q,s)}$ contains N_q. U's computational cost depends on the data structure for representing $N_{A(q,s)}$. The nearest POI N_q from U's location q can be computed in $O(\log k)$ time by locating the cell of Voronoi diagram that contains q when the query response is delivered as the Voronoi diagram $Vor(N_{A(q,s)})$.

Algorithm 1. Spatial cloaking protocol

```
$U$: Generate $A(q, s)$ including $q$ randomly where the size of $A(q, s)$ is greater than $A_{min}$.
$U$: Send $A(q, s)$ to $L$.
$L$: Compute a set of the nearest neighbors $N_{A(q,s)}$ for $A(q, s)$ where $N_{A(q,s)}$ is the set of POIs on the Voronoi cells which intersects with $A(q, s)$.
$L$: Send $N_{A(q,s)}$ to $U$.
$U$: Retrieve the nearest site $N_q$ to $q$ from $N_{A(q,s)}$.
```

4.2 Query Processing

The query processing is based on computation of Voronoi diagram for POIs. We formally define the problem as follows: Given a set $S \overset{def}{\equiv} \{p_1, p_2, ..., p_n\}$ of n distinct points in R^2 and a convex polygon P with m vertices, find a set of the nearest neighbors N_P for P.

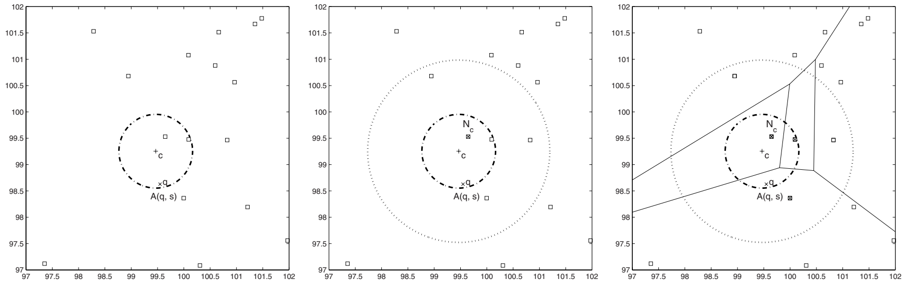
We propose the query processing algorithm using a local Voronoi diagram relevant to the cloaked region since itt is not efficient to maintain the Voronoi diagram for a large entire data set. In particular, for dynamic POIs, the concept of a local Voronoi diagram relevant to the cloaked region is necessarily required since the pre-processed Voronoi diagram is useless when the locations of POIs are dynamically changed. The procedure to compute the intersected Voronoi cells with a polygon P is designed in Algorithm 2.

Algorithm 2. Query processing algorithm

```
Input: a set $S$ of $n$ points, a convex polygon $P$
Output: $N_P$
Find the smallest enclosing disc $D(P)$ for the convex polygon $P$. Let $r$ and $c$ be the radius and the center of $D(P)$, respectively.
Initialize $d$ as $\infty$.
for $s_i \in S$ do
    if $d > dist(c, s_i)$ then
        $d = dist(c, s_i)$
    end if
end for
$r^* = 2 \cdot r + d$
for $s_i \in S$ do
    if $r^* \geq dist(c, s_i)$ then
        Insert $s_i$ into the set of candidate points $S_P$.
    end if
end for
Compute the Voronoi diagram $Vor(S_P)$ for $S_P$.
for $s_i \in S_P$ do
    if a cell $V(s_i) \in Vor(S_P)$ intersects with $P$ then
        Insert $s_i$ into $N_P$.
    end if
end for
return $N_P$
```

Our goal is to identify the minimum set of POIs including the nearest neighbor to the user location. For simple calculation of a threshold r^* for candidate POIs, we use the smallest enclosing disc $D(P)$. The maximum distance d between the enclosing disc $D(P)$ and the nearest POI to the center of $D(P)$ can be used for computing a threshold r^* to choose an adequate set of candidate POIs.

Fig. 3 exemplifies Algorithm 2. Given the user location q and the security parameter s, U constructs a circular cloaked region $A(q, s)$ as the query input (see Figure 2a). L finds a set of candidate POIs for $A(q, s)$ (see Figure 2b) and then compute the local Voronoi diagram for the set (see Figure 2c). Finally, the information about three intersected Voronoi cells with $A(q, s)$ is answered as the query response.



(a) A cloaked region (b) Finding candidate POIs (c) Finding intersections

Fig. 3. A query processing example

Theorem. 1 states that Algorithm 2 is correctly terminated.

Theorem 1. *In Algorithm 2, the nearest POI* N_q *of the query position* q *is necessarily included in* N_P.

Proof. Assume that N_q is not included in N_P. From the assumption, the distance between N_q and the center c of $D(V_P)$ is more than $2{\cdot}r{+}d$. Let N_c be the nearest POI from c. The maximum distance between the enclosing disc $D(P)$ and N_c is $r + d$. Let f be the farthest point on $D(P)$ from N_c.

$$dist(q, N_c) \leq dist(f, N_c) \leq r + d < dist(q, N_q)$$

Therefore N_q is not the nearest POI from q. This result contradicts the assumption.

By Theorem. 1, we can intuitively design the $(r + d)$-approximate algorithm with $O(1)$ communication cost. The nearest POI N_c from the center c is an approximation to the given cloaked region $A(q, s)$.

Theorem 2. *Algorithm 2 runs in* $O(n + t \log t + m)$ *time where* t *is the number of POIs* $\in S_P$.

Proof. We show that Algorithm 2 runs in $O(n + t\log t + m)$ time by analysing the time needed in each step. We start by finding the smallest enclosing disc $D(P)$ for P with m vertices in line 1 which can be solved in $O(m)$ time [17]. Finding the nearest POI N_c from the center c of $D(P)$ in line 3-7 can be solved in $O(n)$ time. Similarly, finding S_P in line 9-13 can be solved in $O(n)$ time. Computing the Voronoi diagram, $Vor(S_P)$, for S_P in line 14 can be solved in $O(t\log t)$ time [24,21,15]. Finding the intersected Voronoi cells with P in line 15-20 can be computed in $O(t + m)$ time [27]. Consequently, the total running time is in $O(n + t\log t + m)$.

The running time of the Algorithm 2 can be improved by using pre-processed data-structures. Finding the nearest POI N_c from the center c of $D(P)$ in line 3-7 can be improved in $O(\log n)$ time. Also, finding S_P in line 9-13 can be improved in $O(\log n+t)$ [2]. In this case, the total running time is in $O(\log n+t\log t+m)$.

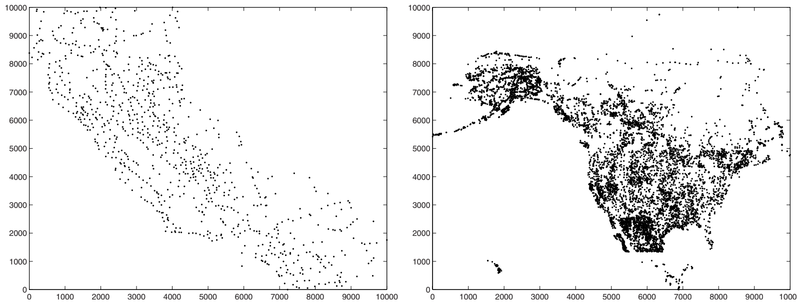
4.3 Approximation Using Sampling

For exact nearest neighbor search, any spatial cloaking techniques including ours may be infeasible depending on scenarios that require extremely high privacy since a larger cloaked region necessarily incurs high communication cost. Considering the constraint of communication cost, the problem can be redefined as follows: Given a constant k_{max}, find a query response with the size which is less than k_{max} for nearest neighbor search. Unfortunately, to achieve this, it is unavoidable to deteriorate accuracy of answer when k_{max} is less than the number of the Voronoi cells which intersect with the cloaked region.

Our strategy is to retrieve POIs according to likelihood. Since the query point q is randomly located at $A(q, s)$, the associated Voronoi cells intersecting the cloaked region $A(q, s)$ as large as possible may be reasonable candidates. This greedy approach guarantees the maximum hit probability of N_q. In Algorithm 2, L computes the size of the intersected area of each Voronoi cell, respectively, and then sorts them in descending order. The first associated k_{max} POIs are answered as the approximate query response. Experimental results have been shown to perform well in practice.

5 Evaluation

In this section, we experimentally evaluate location server L's computational cost, the communication cost and the error distance. The computational cost is measured in terms of the number of POIs computed for the local Voronoi diagram. The communication cost is measured in terms of the number of TCP/IP packets to deliver candidate POIs sent from L back to the user U. We assume that the packet capacity is set to (576-40)/8=67 since a 2D data point takes 8 bytes, a packet has a 40-byte header, and the typical value of a maximum transmission unit (MTU) over a network is 576 bytes [28]. The result error distance is defined as the distance to the candidate nearest neighbor in the query response minus the distance to the actual nearest neighbor.



(a) CA data set

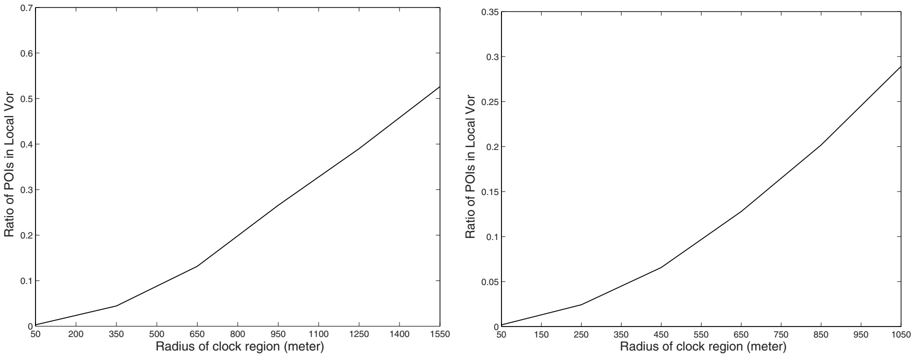
(b) NA data set

Fig. 4. Two data sets

We use the two real datasets: California (CA) with 864 POIs and North-America (NA) with 9,203 POIs (Fig. 4) [26]. The coordinates of points in each dataset are normalized to the square 2D space with extent 10,000 meters. We test the performance by varying the radii of cloaked regions from 50 to 1,550 meters for the CA (from 50 to 1,050 meters for the NA). We generated 100 queries originating at random positions using the Gaussian distribution of the POIs in each dataset.

Regarding the server L's computational cost, we measure the ratio of the POIs that are used to compute the local Voronoi diagram to the entire POIs in each dataset. Fig. 5 shows the relationship between the size of a cloaked region and the number of POIs in a local Voronoi diagram.

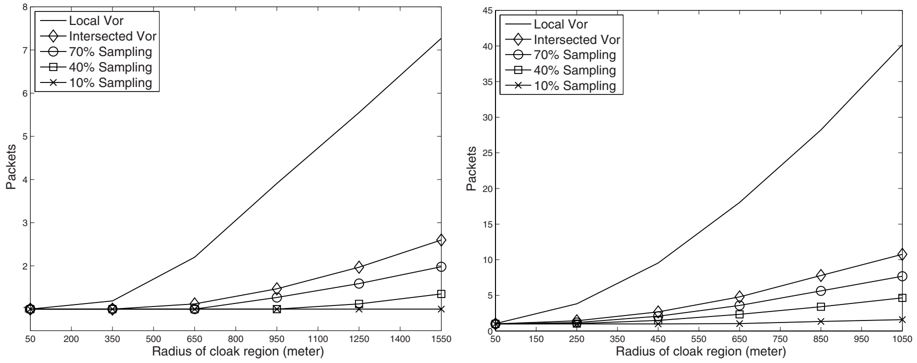
Fig. 6 shows the experimental results for the communication cost. In order to evaluate the performance of our framework, we compare it with a basic spatial cloaking approach (Local Vor in Fig. 6) where all POIs in S_P for local Voronoi diagrams are answered as the query response for nearest neighbor search. For larger cloaked regions, the communication cost of the basic spatial cloaking is



(a) CA

(b) NA

Fig. 5. Size of local Voronoi diagram

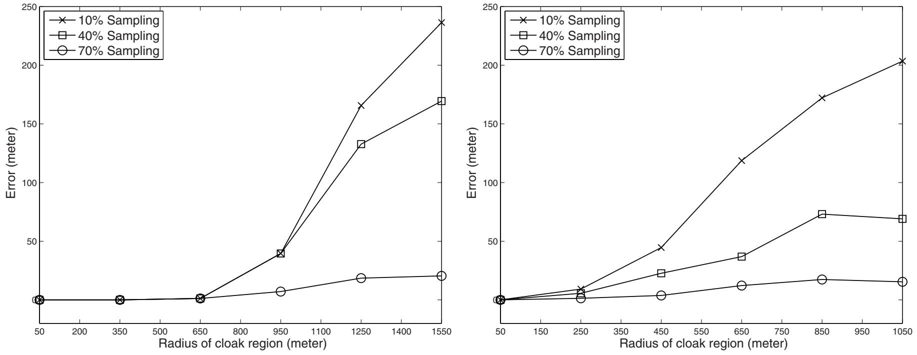


(a) CA (b) NA

Fig. 6. Communication cost

dramatically increasing while the proposed scheme based on the intersected Voronoi cells (Intersected Vor in Fig. 6) is practically acceptable. For example, in CA dataset with sparse POIs, the total communication cost of the proposed scheme is bounded by 3 packets even if the radius of a cloaked region is 1,550 meters.

Fig. 7 shows the measured error distance in sampling methods. Not surprisingly, the communication cost and the result error increases with the size of the cloaked region, respectively. Experimental results show how the proportion of sampling can be set depending on the level of privacy and the communication constraint. In the proposed scheme, a user can achieve the maximum permitted communication cost by controlling the proportion of sampling from intersected Voronoi cells. In these datasets, the sampling of 70% intersected Voronoi cells scales well with the size of the cloaked region. We observe that the sampling method based on Voronoi diagram offer reasonable accuracy and low communication cost.



(a) CA (b) NA

Fig. 7. Distance error

The proposed scheme guarantees that the adversary cannot obtain the information about user location within the cloaked region. The security requirement is achieved defined in Section 3.3 since the size of the cloaked region is greater than A_{min} by the proposed protocol.

6 Conclusions

In this paper, we proposed a spatial cloaking using range search in location-based services for nearest neighbor search. The main idea is to use the adaptive range search query based on Voronoi diagram. In our model, the spatial cloaking problem is interpreted as finding the intersections of Voronoi cells with a cloaked region. We propose a simpler and more flexible protocol based on computation of the Voronoi diagram which we are locally interested. Also, we experimentally investigate the trade-offs between communication/computational cost and levels of privacy (sizes of cloaked regions). Therefore it is applicable in simple client-server architectures since our architecture does not require a trusted middleware. Also, users can flexibly achieve the required communication cost by controlling the proportion of sampling from intersected Voronoi cells.

We will study the extension of the proposed system using the network Voronoi diagram [20] to the road networks with the movement on line segments instead of free-moving since the user's available movement may be restricted by paths such as roads in real applications. Also, one of interesting applications is optimal route planning problem [19,6].

Acknowledgements

The author would like to thank Ross Anderson, Alastair Beresford, and Ji Won Yoon for their careful attention and insightful comments.

References

1. Google latitude (2009)
2. Aggarwal, A., Hansen, M., Leighton, T.: Solving query-retrieval problems by compacting voronoi diagrams. In: STOC 1990: Proceedings of the twenty-second annual ACM symposium on Theory of computing, pp. 331–340. ACM, New York (1990)
3. Chow, C.-Y., Mokbel, M.: Enabling private continuous queries for revealed user locations, pp. 258–275 (2007)
4. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Berlin
5. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
6. Frikken, K.B., Atallah, M.J.: Privacy preserving route planning. In: WPES 2004: Proceedings of the 2004 ACM workshop on Privacy in the electronic society, pp. 8–15. ACM, New York (2004)

7. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)
8. Ghinita, G.: Understanding the privacy-efficiency trade-off in location based queries. In: SPRINGL 2008: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, pp. 1–5. ACM, New York (2008)
9. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private queries in location based services: anonymizers are not necessary. In: SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 121–132. ACM, New York (2008)
10. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: MobiSys 2003: Proceedings of the 1st international conference on Mobile systems, applications and services, pp. 31–42. ACM, New York (2003)
11. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: MobiSys 2004: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pp. 177–189. ACM, New York (2004)
12. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Transactions on Knowledge and Data Engineering 19(12), 1719–1733 (2007)
13. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 239–257. Springer, Heidelberg (2007)
14. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Proceedings. International Conference on Pervasive Services, ICPS 2005, July 2005, pp. 88–97 (2005)
15. Lee, D.: Furthest neighbour voronoi diagrams and applications. Technical Report Report 80-11-FC-04, Dept. Elect. Engrg. Comput. Sci., Northwestern Univ., Evanston, IL (1980)
16. Maria Damiani, C.S.: Elisa Bertino. Probe: an obfuscation system for the protection of sensitive location information in lbs. Technical report
17. Megiddo, N.: Linear-time algorithms for linear programming in r3 and related problems. In: 23rd Annual Symposium on Foundations of Computer Science, 1982. SFCS 2008, November 1982, pp. 329–338 (1982)
18. Mokbel, M.F., Chow, C.-Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proceedings of the 32nd international conference on Very large data bases, pp. 763–774. ACM, New York (2006)
19. Nergiz, M.E., Atzori, M., Saygin, Y.: Towards trajectory anonymization: a generalization-based approach. In: SPRINGL 2008: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, pp. 52–61. ACM, New York (2008)
20. Okabe, A., Boots, B., Sugihara, K., Chi, S.N.: Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd edn. Wiley, Chichester (2000)
21. Preparata, F.P.: Minimum spanning circle. Technical report, Univ. Illinois, Urbana, IL, in: Steps into Computational Geometry (1977)
22. Reed, J., Krizman, K., Woerner, B., Rappaport, T.: An overview of the challenges and progress in meeting the e-911 requirement for location service. IEEE Communications Magazine 36(4), 30–37 (1998)

23. Samarati, P., Sweeney, L.: Generalizing data to provide anonymity when disclosing information (abstract). In: PODS 1998: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, p. 188. ACM, New York (1998)
24. Shamos, M.I., Hoey, D.: Closest-point problems. In: SFCS 1975: Proceedings of the 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), Washington, DC, USA, pp. 151–162. IEEE Computer Society, Los Alamitos (1975)
25. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10(5), 557–570 (2002)
26. Theodoridis, Y.: The r-tree-portal (2009)
27. Toussaint, G.T.: A simple linear algorithm for intersecting convex polygons. The Visual Computer 1, 118–123 (1985)
28. Yiu, M.L., Jensen, C., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: IEEE 24th International Conference on Data Engineering, ICDE 2008, April 2008, pp. 366–375 (2008)

Visualizing Privacy Implications of Access Control Policies in Social Network Systems

Mohd Anwar[1], Philip W.L. Fong[1], Xue-Dong Yang[2], and Howard Hamilton[2]

[1] Department of Computer Science, University of Calgary, Alberta, Canada
{manwar,pwlfong}@ucalgary.ca
[2] Department of Computer Science, University of Regina, Saskatchewan, Canada
{yang,hamilton}@cs.uregina.ca

Abstract. We hypothesize that, in a Facebook-style social network system, proper visualization of one's extended neighbourhood could help the user understand the privacy implications of her access control policies. However, an unrestricted view of one's extended neighbourhood may compromise the privacy of others. To address this dilemma, we propose a privacy-enhanced visualization tool, which approximates the extended neighbourhood of a user in such a way that policy assessment can still be conducted in a meaningful manner, while the privacy of other users is preserved.

1 Introduction

One of the main purposes of privacy preservation is impression management [1,2]. This is particularly true in the context of social network systems. A profile owner selectively grants a profile viewer access to her profile items in accordance with the impression she wants to convey. For example, say Jill is a friend of Alice, and Bob is a friend of Jill. For proper impression management, Alice may grant Jill, but not Bob, access to her sorority photo album. To check whether her policy allows her to convey the desired impression, Alice may want to look at her profile from the lenses of Bob and Jill, to find out what Bob as well as Jill can see. In our everyday life, we look into a mirror to get a sense of what others see when they look at us. We use the term ***reflective policy assessment*** to refer to this process of assuming the position of a potential accessor for the sake of assessing the privacy implications of access control policies.

Authorization in a social network system is primarily based on the topology of the social graph, which is co-constructed by all the users of the system. It is therefore difficult for a user to mentally keep track of the topology of her constantly changing social network. Furthermore, one's needs for privacy is constantly changing, requiring a user to constantly perform policy assessment. As a result, reflective policy assessment is a nontrivial undertaking. Tool support is definitely desirable.

Unfortunately, a privacy dilemma is inherent in reflective policy assessment. To assess policies reflectively, a user must begin with identifying a potential accessor who is of interest to her. This, however, could lead to breaching the privacy of the potential accessor, as the latter may not want her identity to be disclosed to the user conducting the policy assessment. Suppose the running example is situated in Facebook. If Bob

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 106–120, 2010.

adopts a privacy setting that allows his identity to be revealed only to friends but not friends of friends, then Alice will not be able to conduct reflective policy assessment against Bob without breaching his privacy.

This privacy dilemma is not specific to just Facebook. Fong et al. proposed an access control model to delineate the design space of privacy preservation mechanisms in Facebook-style social network systems [3]. In this model, policies such as “only friends” and “friends of friends” are but examples of more general ***topology-based policies***, whereby accessibility is determined by the present topology of the social graph. For example, Alice may adopt the policy that grants access to her sorority photo album only if the accessor shares three common friends with her. With these policies, it would even be more important to have access to one’s extended neighbourhood in addition to her immediate friends for the purpose of policy assessment.

This dilemma is rooted in the asymmetric nature of trust. In the process of reflective policy assessment, a resource owner (e.g., Alice) conceptualizes the level of trust she is willing to invest in a potential accessor (e.g., Bob). Yet, this endeavor is possible only if the identity of the potential accessor is known to the resource owner, the feasibility of which may not always be possible because the potential accessor may not trust the resource owner.

This paper is about the design of a privacy enhanced visualization tool for Facebook-style social network systems (FSNSs) to facilitate reflective policy assessment while preserving the privacy of potential accessors. The visualization tool helps a user assess her access control policies by: (a) visually depicting the extended neighbourhood of her social graph and (b) allowing her to inspect her profile from the view point of a potential accessor at her extended neighbourhood. Our contributions are the following:

1. We introduce the notion of reflective policy assessment, which helps a user assess the privacy implications of her policies by positioning herself as a potential accessor. We also discover and address an inherent privacy dilemma of reflective policy assessment.
2. We translate the concept of reflective policy assessment into a concrete visualization tool for policy assessment. Since this tool would not require the knowledge of access control policies of all the users of the system, it can be implemented on the client side (e.g., as a third-party Facebook application).
3. At the core of our visualization technique is a visual representation of a user’s extended neighbourhood. We establish graph-theoretic properties common to the social graphs of FSNSs. Based on these properties, we devise an algorithm to generate a surrogate of a user’s extended neighbourhood. This surrogate can be examined for reflective policy assessment without violating the privacy of other users.

The organization of this paper is as follows. Sect. 2 describes an access control model for FSNSs. In Sect. 3, we present the main idea of assessing policies through visualization. In Sect. 4, we present an algorithm for generating a surrogate of a user’s extended neighbourhood for policy assessment. Sect. 5 discusses subtle issues in our visualization approach. Sect. 6 presents some open questions on how to evaluate the proposed visualization technique. Sect. 7 surveys related literature, and Sect. 8 describes conclusion and future work.

2 An Access Control Model for FSNSs

In this work, we study reflective policy assessment for the family of FSNSs proposed in [3]. Specifically, [3] defines an access control model for social network systems, of which Facebook is but one instantiation. The model generalizes the authorization scheme of Facebook, allowing a more expressive policy vocabulary (see below). We argued in [3, Sect. 5] that careful instantiations of the model can serve as the access control infrastructure of information sharing systems. This section briefly outlines the FSNS access control model so as to anchor the discussion in the sequel. Formal details of this model can be found in [3].

Profile and Profile Items. An FSNS allows each user to construct a representation of him- or herself in the form of a ***profile***. A profile displays such ***profile items*** as personal information, multimedia contents, activity logs, or other user-authored contents. Users may grant one another access to their profile items.

Search Listings. Access to profile items is authorized in two stages (See Fig. 1). In ***Stage I***, the accessor must ***reach*** the ***search listing*** of the profile owner. Then in ***Stage II***, the accessor requests access to the profile, and profile items are selectively displayed. The search listing of a user could be seen as a "capability" [4,5] of the user in the system, through which access is mediated. There are two means by which a profile can be reached in Stage I: ***global name search*** and ***social graph traversal***.

Global Name Search. The first means to reach a search listing is to conduct a global name search. A successful search would produce for the accessor the search listing of the target user. A profile owner may specify a ***search policy*** to allow only a subset of users to be able to reach her search listing through a global name search.

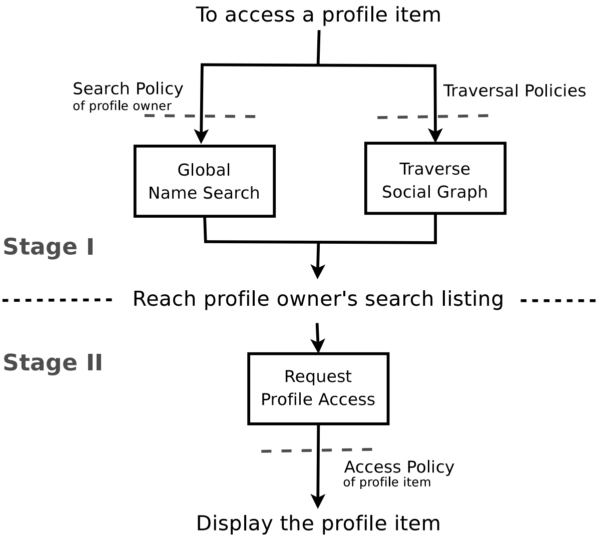


Fig. 1. Authorization procedure for FSNSs

Social Graph Traversal. A second means to reach a search listing is by traversing the ***social graph***. Users can articulate their relationships with one another through the construction of ***friend lists***. Every user may specify a set of other users as her ***friends***. This induces a simple graph in which users are nodes and relationships are edges. A user may traverse this graph by examining the friend lists of other users. More specifically, the friend list of a user is essentially the set of search listings of her friends. A user may restrict traversal by specifying a ***traversal policy***, which specifies the set of users who are allowed to examine her friend list once her search listing is reached.

Profile Access. Once the search listing of a profile owner is reached, the accessor may choose to access the profile, and thereby, initiate Stage II of authorization. Since a profile owner may assign an ***access policy*** to each profile item, not every accessor sees the same profile items when a profile is accessed.

Friendship Articulation. Articulating friendship involves a consent protocol, whereby users interact with one another via a fixed set of ***communication primitives*** (e.g., friendship invitation, accepting an invitation, etc). Once a mutual consent is reached, that friendship is recognized by the FSNS. When a sender initiates a communication primitive against a receiver, the search listing of the latter must be reached before the communication primitive can be initiated. A user can prevent others from initiating a certain communication primitive against her by assigning a ***communication policy*** to that primitive.

Topology-Based Policies. User activities are controlled by user-specified policies (i.e., search, traversal, access and communication policies). Each FSNS offers a fixed policy vocabulary, so that users may adopt policies from the vocabulary to identify sets of privileged users. Since there is no global name space of users, these predefined policies identify user sets by an intensional specification[1]. For example, one may specify that a certain profile item is accessible only by members of the "University of Calgary" network. In [3], we examined a family of intensionally-specified policies known as ***topology-based policies***, which identify privileged users solely in terms of the current topology of the social graph. For instance, one may mandate that a certain profile item is visible only to "friends of friends". We proposed in [3] a number of topology-based policies that are not currently supported by Facebook, but nevertheless possess rich social significance. A sample of these topology-based policies are shown in Fig. 2.

Policy predicate:	*When is access allowed*
distance$_k$:	distance between owner and accessor is no more than k
clique$_k$:	owner and accessor belong to the same k-clique (i.e., they belong to the same close-knit group)
common-friends$_k$:	owner and accessor share k common friends (i.e., accessor is a known quantity)

Fig. 2. A sample of topology-based policies

[1] An extensional definition specifies a set by enumerating its members (e.g., $S = \{0, 1, 2\}$). An intensional definition specifies a set by stating the characteristic property of its members (e.g., $S = \{x \in \mathbb{N} \mid x < 3\}$).

As we mentioned before, it is cognitively challenging for an FSNS user to understand the privacy implications of adopting a certain topology-based policy. The next section presents a visualization technique that supports reflective policy assessment in the presence of topology-based policies.

3 A Privacy-Enhanced Visualization Technique

A Mirror-based Visualization Technique. Our visualization technique seeks to provide a mirror-like affordance to users in FSNSs. To create a desired impression, we repeatedly look into the mirror and adjust our getup until we are satisfied. A mirror allows us to see what others see when they look at us. The process of formulating access control policies is similar to what it takes to create a desired look. With our ever changing social network and ever changing desire for privacy, a user needs to repeatedly assess and adjust their policies. We propose a mirror-like tool to help a user visualize what others see when they look at her.

Our proposed visualization tool offers the following functionalities to a profile owner.

1. The tool provides a visual representation of an extended neighbourhood of a profile owner in the social graph. The profile owner may specify the size of her extended neighbourhood.
2. This tool allows the profile owner to point to any user in the extended neighbourhood as a potential accessor of her profile. This action signals to the tool that the profile owner intends to position herself as the selected user and examine her profile from the view point of that user.
3. The tool displays a succinct representation of the profile, as seen from the eyes of the potential accessor.
4. The tool suggests potential accessors representing interesting access scenarios (see Sect. 5.2).

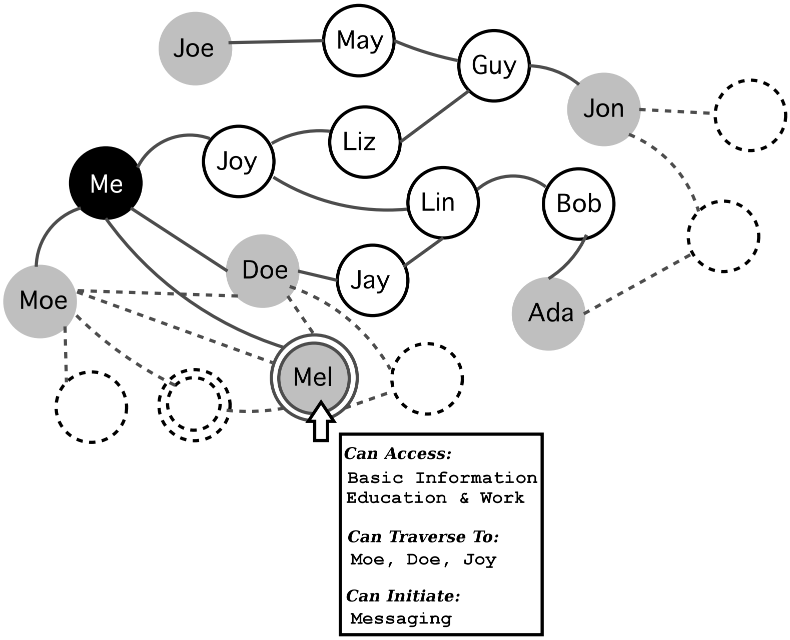
This tool contributes to policy assessment in the following ways:

What-if Analysis: It allows a profile owner to perform "what-if" analysis on her access policies. More specifically, it allows her to assess the adequacy of her access policies in concrete access scenarios, and to evaluate the effect of adopting these policies when her extended neighbourhood possess a certain topological structure.

Targeted Effort: As the tool displays how other users are topologically related to a profile owner, it helps her identify topologically interesting nodes in the extended neighbourhood, thereby allowing her to properly target her policy assessment effort. For example, in Fig. 4, the node *FOF* corresponds to an interesting access scenario when the profile owner *Me* attempts to assess a "friends of friends" policy.

Visualizing without Breaching Privacy. The visual representation of the extended neighbourhood must be generated in such a way that the privacy of a potential accessor is preserved. To see this, recall in Sect. 2 that not every potential accessor is reachable from the profile owner, even if there is a path between them. This scenario may arise if at least one of the intermediate nodes along the path has a traversal policy that prevents the profile owner from examining the friend list of that intermediate node. Consequently, depicting the extended neighbourhood in full accuracy compromises privacy. Fortunately, an accurate rendering of the extended neighbourhood is not necessary for reflective policy assessment. Rather, an approximate rendering that exhibits the topology typical of social networks should suffice. Therefore our approach is to approximate the unreachable region of the extended neighbourhood by generating synthetic nodes and edges in a way that preserves such properties of social networks as power law vertex degree distribution [6] and small-world characteristic [7]. Details of the graph generation algorithm can be found in Sect. 4.

Mockup. In Fig. 3, we show a mockup of our visualization tool. Here, the black node is the profile owner (*Me*). White nodes (e.g., *Jay*) and solid edges (e.g., *Jay-Doe*) depict the interior of the profile owner's reachable region in the social graph. Grey nodes (e.g., *Doe*) mark the boundary (inclusive) of the reachable region. The dotted nodes and dotted edges are generated to approximate the unreachable region of the profile owner. The double-circled dotted or solid nodes are the potential accessors representing interesting access scenarios (as suggested by our tool, see Sect. 5.2 for details). As the profile owner selects a potential accessor by pointing her cursor over the latter, an information box pops up. The information box displays what profile items of the profile owner that the selected user can see as a result of the profile owner's current policies. Specifically, the information box displays three categories of information: (i) the profile



Can Access:
Basic Information
Education & Work

Can Traverse To:
Moe, Doe, Joy

Can Initiate:
Messaging

Fig. 3. A prototypical visualization tool to facilitate reflective policy assessment. The black node represents the profile owner. The double-circled node depicts a potential accessor representing an interesting access scenario. When the profile owner points to the potential accessor, Mel, the pop-up box displays a configuration of the profile owner's profile that Mel sees.

items of the profile owner that the selected user can access, (ii) a list of the profile owner's friends that the selected user can reach through the profile owner, and (iii) a list of communication primitives that the selected user can initiate against the profile owner.

Section (i) of the information box is a "reflection" of the profile under assessment. This section supports the assessment of access policies. Section (ii) of the information box supports the assessment of traversal policies. A user's traversal policy has privacy implications not only on the user, but also on her friends. Specifically, an overly relaxed traversal policy will expose one's friends to unwanted accessors. In a similar vein, section (iii) of the information box supports the assessment of one's communication policies.

As an example, in Fig. 3, when the profile owner *Me* points to *Mel*, the tool displays the following: (i) *Mel* can access two profile items of the profile owner: "*Basic Information*" and "*Education and Work*"; (ii) *Mel* can reach *Moe*, *Doe* and *Joe* through *Me*; (iii) *Mel* can send a message to *Me*, but cannot invite *Me* to be a friend.

Assessing Topology-based Policies. A critical reader may question why it is necessary to consider unreachable nodes in the process of reflective policy assessment. We illustrate the utility of this practice by giving some examples. Consider the extended neighbourhood of user *Me* in Fig. 4. We show how various topology-based policies need to be evaluated from the view point of unreachable nodes.

distance$_k$: Suppose user *Me* adopts distance$_5$ as the access policy for her wedding video, thereby granting access to anyone within a distance of five. Let us suppose further that *Jon* is at distance four, whose traversal policy does not allow *Me* to traverse to *Jon*'s friends, including, for example, *D5*. However, user *Me* may precisely want to examine her profile from the perspective of *D5*, which is at distance five from *Me*, in order to evaluate her distance$_5$ policy.

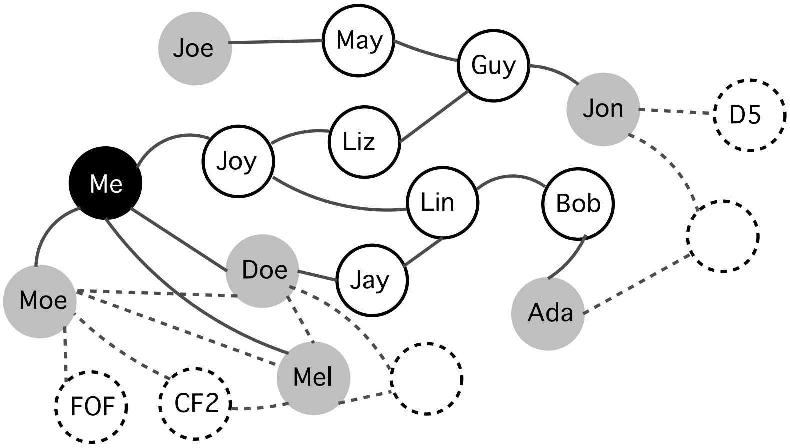


Fig. 4. A visual representation of a profile owner's extended neighbourhood. The black node depicts the profile owner. The grey nodes mark the boundary of the reachable region. The dotted nodes and dotted edges depict the unreachable region of the profile owner's extended neighbourhood.

common-friends$_k$: Suppose the profile owner *Me* specifies common-friends$_3$ as the access policy of her “Contact Information”, so that the latter is accessible to those users sharing three common friends with *Me*. According to Fig. 4, users *Me* and *CF2* have only two common friends (*Moe* and *Mel*). It is to the interest of user *Me* to assess her policies reflectively from node *CF2*. Lets suppose Moe and Mel do not allow someone to look at their friends list. Therefore, rendering the node *CF2* would be a breach of *Moe*’s and *Mel*’s privacy. Furthermore, by breaching *Moe*’s and *Mel*’s traversal policy, *CF2*’s privacy is also breached since *CF2* delegates its reachability to *Moe* and *Mel*.

clique$_k$: Suppose user *Me* specifies an access policy, clique$_4$, for her “Status”. That is, access is granted to her friends who belong to the same 4-clique as she does. In Fig. 4, users *Me*, *Moe*, *Doe* and *Mel* belong to the same 4-clique. Even though user *Me* needs to confirm that *Moe* and *Doe*, *Doe* and *Mel*, and *Mel* and *Moe* are friends in order to assess her clique$_4$ policy, the traversal policies of *Doe*, *Moe* and *Mel* do not allow the *Me* to discover these relationships.

4 Constructing a Social Graph for Policy Assessment

This section describes an algorithm for generating a visual representation of the social graph for policy assessment. We set the stage by describing some graph-theoretic properties of FSNS social graphs (Sect. 4.1), and then apply the properties to devise the algorithm and establish its correctness (Sect. 4.2).

4.1 Properties of Social Graphs

A node v is u-traversable if the traversal policy of v allows u to examine the friend list of v. If there is a uv-path $uv_1 \dots v_n v$ in the social graph such that every v_i is u-traversable, then we say user v is ***u-reachable***. Otherwise, v is ***u-unreachable***. A u-reachable node is a ***u-interior node*** if it is u-traversable, and a ***u-fringe node*** otherwise. An edge is ***u-visible*** if one of its ends is a u-interior node, otherwise it is ***u-hidden***. The node u in the above definitions is called the ***origin***. We drop the “u-” prefix when the origin is clear from the context.

Property 1. Given an origin, every neighbour of an interior node is reachable, and thus, no hidden edge can have an interior node as an end.

Property 2. Suppose an origin is given. By definition, at least one end of each visible edge is an interior node. Therefore, no visible edge can join two fringe nodes.

4.2 A Graph Generation Algorithm

We present an algorithm for generating a graph to approximate an extended neighbourhood of a user u in the social graph. The generated graph is composed of two regions. The first region is made up of the reachable nodes and the visible edges. The second region is randomly generated to approximate the unreachable nodes and the hidden edges of the social graph. To ensure that the randomly generated region reflects the topological structure of a typical social graph, we employ the R-MAT [8] algorithm, which randomly generates graphs exhibiting statistical properties of a real-world social network. (Other appropriate graph generation algorithms can also be used.)

```
Algorithm. $A(u, M, N)$
1. Using $u$ as the origin, construct a graph consisting of all reachable nodes and visible edges.
2. Temporarily remove all interior nodes and visible edges, leaving only the fringe nodes.
3. Add $N$ “synthetic nodes”.
4. Use R-MAT to randomly generate $M$ “synthetic edges”.
5. Add back the interior nodes and visible edges removed in step 2, and return the resulting graph.
```

Algorithm A has three parameters: the origin u, the number N of synthetic nodes to be generated, and the number M of synthetic edges to be added into the graph. We plan to decide on the default value of N and M heuristically based on our forthcoming user study. Step 1 can be achieved by an elementary third-party Facebook application that performs a breadth-first search[2]. This means the algorithm can be executed on the client side, execising no more privileges than the profile owner conducting policy assessment.

The correctness of algorithm A on generating an approximated extended neighbourhood hinges on two conditions. The first correctness condition is that, because synthetic edges are surrogates of hidden edges, the former should only be generated where the latter may occur. By **Property 1**, no hidden edge can have an interior node as an end, and thus synthetic edges should only be generated among fringe nodes and sythetic nodes. This is guaranteed by the removal of interior nodes from consideration in Step 2.

A second correctness condition is that the invocation of R-MAT in Step 4 must begin with an empty graph, so that the statistical properties of R-MAT is preserved. (This condition is not specific to R-MAT, and is necessary even if other graph generation algorithms are used.) By **Property 2**, no visible edge can join two fringe nodes, and thus Step 4 always starts with an empty graph.

5 Issues and Discussion

5.1 Information Leakage

Displaying the profile of a user from the perspective of an accessor may allow the profile owner to infer information about the accessor that is otherwise inaccessible, thereby violating the privacy of the latter. To make the objection concrete, consider the following “attack”: Suppose a user u imposes an access policy on a certain profile item o, so that o is visible to someone who belongs to the “University of Calgary” network. Suppose further that user v is a member of that network, but she sets up her access

[2] For example, the third-party Facebook application TouchGraph performs a similar search.

policies so that this fact is not accessible by u. Now, by performing reflective policy assessment from the view point of v, and observing that o is visible to v, u can infer that v belongs to the said network. Thus the privacy of v is breached.

It turns out that information leakage can be prevented by adopting topology-based policies (see Sect. 2 or [3, Sect. 4.2]), so that reflective policy assessment does not leak information that is not already accessible by the user conducting the assessment. With topology-based policies, accessibility is determined solely by the current topology of the social graph. For example, the policies in Fig. 2 are all topology-based. If the FSNS offers only topology-based policies in its policy vocabulary, then mirror-based visualization reveals no other information than the current topology of the social graph. The question then is, does reflective policy assessment disclose topological information that a user does not already possess? The answer is negative. Recall in Sect. 4 that visible edges are already accessible by the profile owner. Hidden edges do not take part in reflective policy assessment. Instead synthesized edges are randomly generated surrogates of hidden edges in reflective policy assessment. Therefore, the topological information that is revealed by reflective policy assessment is either already available (visible edges) or anonymized (synthesized edges). Topological information induced by hidden edges is not revealed at all.

5.2 Recommending Access Scenarios

A feature of our visualization technique is to recommend nodes (potential accessors) that represent interesting access scenarios by highlighting such nodes so that a profile owner can target her policy assessment effort against these potential accessors. In the following we elaborate on what we mean by "interesting access scenarios", and provide additional justifications of our approach.

Once the visualization tool has generated an extended neighbourhood of the profile owner, some nodes are indistinguishable from an access control point of view. More specifically, the appearances of the owner's profile as accessible from the view points of these nodes may be identical. Consequently, there is no need for the profile owner to conduct reflective policy assessment against more than one of these nodes. In short, the various profile appearances partition the nodes into equivalence classes. Each equivalent class represents a distinct ***access scenario***. An access scenario is interesting if it has not been encountered before.

If k distinct topology-based policies are assigned to the items in the profile[3], then there is at most 2^k distinct profile appearances, and thus the same number of distinct access scenarios. To see this, note that what induces a specific profile appearance is the satisfiability of the k policy predicates when a node is given. Consider the following example. Suppose P_1 and P_2 are the policy predicates assigned to the various profile items. Two nodes that both satisfy P_1 but violate P_2 are going to produce the same profile appearance, and thus belong to the same access scenario[4].

[3] The same policy can be assigned to multiple profile items, while certain policies in the policy vocabulary may not be assigned to any profile item at all. Therefore, we do not concern ourselves with the number of profile items or the size of the profile vocabulary.

[4] Note that 2^k is only an upper bound, because some profile appearances are not feasible. For example, no node can violate $\mathsf{distance}_2$ but satisfy clique_4.

Therefore, a tool that supports reflective policy assessment should: (a) help the profile owner identify an enough number of distinct access scenarios (cf. Sect. 6.3) so that the profile owner can have confidence of its privacy settings, and (b) provide a means to describe the individual access scenarios to the profile owner. We intend our visualization tool to track the access scenarios that the profile owner has encountered within a policy assessment session. The tool will selectively highlight a node if it corresponds to a novel access scenario. Multiple extended neighbourhoods can then be generated to help cover commonly occurring access scenarios. Requirement (a) is thus addressed. We also anticipate that the visual depiction of the extended neighbourhood provides an efficient and comprehensible description of access scenarios, thereby addressing requirement (b).

It may appear that since we already know the 2^k access scenarios, there is no need to randomly sample extended neighbourhoods of the profile owner. All we need is to enumerate the 2^k access scenarios, and display the corresponding profile appearances. Unfortunately, this hypothetical solution does not address requirement (b). Recall that a description of the access scenario must be conveyed to the profile owner. We believe that a visual depiction is more effective than a verbal summary of potential access scenarios, such as "$\mathsf{common\text{-}friends}_4$ but not clique_4". Now the question arises whether we should systematically construct visual representation of access scenarios for the policy space of the profile owner. We desire our tool to be indifferent to the specific choice of policy vocabulary. If we are to enumerate all access scenarios, our tool has to do an exhaustive search in the space of all possible social graphs, resulting in exponential time complexity. Instead, our approach of randomly generating the extended neighbourhood can be seen as a Monte Carlo strategy to cope with the intractability of enumerating arbitrary graph-theoretic access scenarios.

6 Open Questions

Our proposal motivates a number of open questions.

6.1 To What Extent Does Our Visualization Technique Facilitate the Assessment of Access Control Policies in FSNSs?

If a tool is effective in supporting policy assessment, we should observe that privacy-aware users tend to formulate a different set of policies after adopting the tool. An empirical user study will help us test if this is indeed the case for our visualization technique. Such a user study shall compare the policies formulated by the user in at least three configurations: (i) no visualization is available, (ii) mirror-based visualization with the rendering of reachable nodes only, (ii) mirror-based visualization with the rendering of both reachable and unreachable nodes.

6.2 How Do We Build a Testbed to Run the Proposed User Study?

A deployed FSNS, such as Facebook, would have been a convenient environment to conduct the proposed user study. There are, however, two problems with this approach.

First, not all topology-based policies are supported in Facebook. As a result, the effectiveness of reflective policy assessment against advanced topology-based policies cannot be gauged. Second, such a study will harvest information of users located in the reachable region of a participant. This setup thus requires consent from a population much larger than the participating group. Even if this aggressive experimental design is approved by the institutional research ethics committee, successfully obtaining consent from such a large population is not likely. We anticipate that the resolution of this problem will involve a clever design of a simulated environment that addresses these privacy challenges.

6.3 To What Extent Are the Randomly Generated Graphs (Sect. 4.2) Useful Approximations of the Unreachable Region of One's Extended Neighbourhood?

We hypothesize that the graphs generated by algorithm A cover topologically interesting access scenarios needed by the profile owner for conducting reflective policy assessment against unreachable nodes. Intuitively, repeated policy assessment on multiple generated graphs should increase the coverage of topologically interesting access scenarios. A natural research question is thus the following: "*how many graphs does one need to generate in order to gain enough confidence on the policies under assessment?*" A probabilistic analysis of this problem is in order.

6.4 How Well Does Our Visualization Tool Facilitate Reflective Policy Assessment in a Very Large Extended Neighbourhood?

It would be burdensome for a user with a very large extended neighbourhood to assess her policies against every access scenario. However the profile owner needs not conduct reflective policy assessment on every node since some of these nodes have the same access privilege to the profile. Our tool groups access scenarios into equivalent classess, and thereby, suggests a distinct access scenario per equivalent class. Additionally, we can apply *focus + context* technique on a hyperbolic plane [9] to effectively render a large neighbourhhod for reflective policy assessment. Using this technique, we want to assign more display space to some interesting access scenarios (to render greater focus), while still embedding the focused access scenarios into the context of entire neighbourhood. A profile owner can easily move her mouse pointer to focus on a different part of her extended neighbourhood and perform policy assessment against different access scenarios.

7 Related Works

Assessing the security implications of access control policies traditionally lies in the domain of safety analysis [10,11], or, more recently, security analysis [12,13]. When the projection of security implications becomes a challenging computational problem, safety or security analyses are indispensable. While appreciating the scope and analytical rigor of such approaches, this paper seeks to address the *cognitive challenges* of users in the projection of the *privacy implications* of their access control policies. A visualization tool can reduce the cognitive load of users in policy assessment. It is also a better fit with the requirements of impression management.

Visualization techniques have long been used in social network analysis [14]. With the soaring popularity of online social networks, visualization techniques are widely used to empower users of such networks. For example, Heer & boyd employed visualization techniques for exploration and navigation of large-scale online social networks [15]. Facebook offers a profile owner to see how a friend sees her profile[5]. Reeder et al. proposed a visualization technique to support authoring of security policies [16], whereby the access control matrix is rendered as an expandable grid representation. Ours and Reeder et al.'s work share a common underpinning of visualizing authorization decisions under the assumption of some security policies. In our work, when a profile is displayed from the view point of a potential accessor, we are essentially rendering a segment of the row in access control matrix corresponding to that accessor. Our work is distinct from their work on two counts: (i) our work is tailored for the assessment of topology-based access control policies in the context of social network systems, and (ii) we are concerned with preserving the privacy of potential accessors. To the best of our knowledge, we are the first to propose visualization of social network for access control policy assessment. Our proposed visualization technique supports impression management for a family of FSNSs. This family was defined by Fong et al. [3], who formally specify an access control model that delineates the design space of social network systems employing the same access control paradigm as Facebook. A distinctive feature of FSNSs is that no global name space is available for identifying users, and thus access control policies are specified in terms of the present topology of the social graph. This element of distributed access control causes policy assessment to be a nontrivial undertaking, thereby necessitating our visualization technique. Furthermore, Fong et al. formulated some policies that are purely based on topological information: e.g., Degree of Separation, Known Quantity, Clique, etc.

A number of recent proposals attempt to advance beyond the access control mechanisms found in commercial social network systems. A notable example is that of Carminati et al., in which a decentralized social network system with relationship types, trust metrics and degree-of-separation policies is developed [17,18,19,20,21]. An interesting research issue is to design tools that support reflective policy assessment in these next-generation social network systems.

8 Conclusion and Future Work

We anticipate that our visualization technique can reduce users' cognitive load in understanding the privacy implications of their access control policies in a FSNS. Specifically, this visualization technique helps a profile owner assess her policies by displaying how potential accessors are topologically related to her in an extended neighbourhood, and allowing her to visually assess her policies via a mirror-like facility from the perspective of a potential accessor of her choice. This technique supports the reflective assessment

[5] http://www.facebook.com/privacy/?view=profile

of access, traversal and communication policies in FSNSs. We plan to conduct an empirical user study to gauge the effectiveness of this visualization technique. We also plan to address the theoritical question of figuring out the number of graph samples needed for inducing confidence on reflective policy assessment.

References

1. Goffman, E.: The Presentation of Self in Everyday Life. Anchor-Doubleday, New York (1961)
2. Patil, S., Kobsa, A.: Privacy as impression management. Technical Report UCI-ISR-03-13, Institute for Software Research, University of California - Irvine, Irvine, CA, USA (December 2003)
3. Fong, P.W.L., Anwar, M., Zhao, Z.: A privacy preservation model for Facebook-style social network systems. In: Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS 2009), Saint Malo, France (September 2009)
4. Dennis, J.B., van Horn, E.C.: Programming semantics for multiprogrammed computations. Communications of the ACM 9(3), 143–155 (1966)
5. Miller, M.S., Yee, K.P., Shapiro, J.: Capability myths demolished. Technical Report SRL2003-02, System Research Lab, Department of Computer Science, The John Hopkins University, Baltimore, Maryland, USA (2003)
6. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: Proceedings of ACM Special Interest Group on Data Communications (SIGCOMM 1999), pp. 251–262 (1999)
7. Milgram, S.: The small world problem. Psychology Today 1, 60–67 (1967)
8. Chakrabarti, D., Faloutsos, C., Zhan, Y.: Visualization of large networks with min-cut plots, A-plots and R-MAT. International Journal of Human-Computer Studies 65, 434–445 (2007)
9. Lamping, J., Rao, R.: The hyperbolic browser: A focus+context technique for visualizing large hierarchies. Journal of Visual Languages and Computing 7(1), 33–35 (1996)
10. Harrison, M.A., Ruzzo, W.L., Ullman, J.D.: Protection in operating systems. Communications of the ACM 19(8), 461–471 (1976)
11. Lipton, R.J., Snyder, L.: A linear time algorithm for deciding subject security. Journal of the ACM 24(3), 455–464 (1977)
12. Li, N., Winsborough, W.H., Mitchell, J.C.: Beyond proof-of-compliance: Safety and availability analysis in trust management. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp. 123–139 (2003)
13. Li, N., Tripunitara, M.V.: Security analysis in role-based access control. In: Ninth ACM Symposium on Access Control Models and Technologies (SACMAT 2004), pp. 126–135 (2004)
14. Freeman, L.C.: Visualizing social networks. Journal of Social Structure 1(1) (2000)
15. Heer, J., boyd, d.: Vizster: Visualizing online social networks. In: Proceeding of IEEE Symposium on Information Visualization, pp. 33–40 (2005)
16. Reeder, R.W., Bauer, L., Cranor, L.F., Reiter, M.K., Bacon, K., How, K., Strong, H.: Expandable grids for visualizing and authoring computer security policies. In: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI 2008), pp. 1473–1482. ACM, New York (2008)
17. Carminati, B., Ferrari, E., Perego, A.: Rule-based access control for social networks. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1734–1744. Springer, Heidelberg (2006)

18. Carminati, B., Ferrari, E., Perego, A.: Private relationships in social networks. In: Proceedings of Workshops in Conjunction with the International Conference on Data Engineering – ICDE 2007, Istanbul, Turkey, pp. 163–171. Springer, Heidelberg (2007)
19. Carminati, B., Ferrari, E.: Privacy-aware collaborative access control in web-based social networks. In: Atluri, V. (ed.) DAS 2008. LNCS, vol. 5094, pp. 81–96. Springer, Heidelberg (2008)
20. Carminati, B., Ferrari, E., Perego, A.: Enforcing access control in web-based social networks. ACM Transactions on Information and System Security (to appear, 2009)
21. Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: A semantic web based framework for social network access control. In: SACMAT 2009: Proceedings of the 14th ACM symposium on Access control models and technologies, pp. 177–186. ACM, New York (2009)

Contextual Privacy Management in Extended Role Based Access Control Model

Nabil Ajam, Nora Cuppens-Boulahia, and Fréderic Cuppens

Institut Télécom
Télécom Bretagne, 2 rue de la Chataîgneraie Cesson-Sévigné 35576
LUSSI Department
{nabil.ajam,nora.cuppens,frederic.cuppens}@telecom-bretagne.eu

Abstract. Typically, informational privacy aims to protect personal data from unauthorized access. In this paper, we propose to use the OrBAC model enhanced by some concepts to model privacy policies. We will take into account the concepts of consent, accuracy, purposes of the access and provisional obligation within role-based access control model.

First, we focus on modelling of the requirement of the data owner consent before delivering the sensitive data. The subscriber defines that he must be notified before terminating the access. The access is delayed until the satisfaction of this condition.

On the other hand, the accuracy of the sensitive data is usually underestimated within privacy models. We design an object hierarchy based on predefined accuracy levels. For this, we propose a derivation rule of sensitive objects. So, data owner can define authorisations based on different object accuracies.

Furthermore, access control models usually permit the access to the stored data based on the role of the requester. We propose to extend this concept to take into account the purpose of the access. For this, we take advantage of the OrBAC user-declared context.

Finally, we propose in this work to model the provisional obligations after accessing personal information. Third parties must notify data controller about further usage over collected data.

To validate our approach, we show how the resulting model can be used to model the privacy policy for a location-based service. This can be applied within a mobile operator organization.

Keywords: Privacy, privacy model, access control model, consent, obligation, purpose, accuracy.

1 Introduction

Since Information Technologies require to collect, store and disclose private information about individuals, privacy concerns are increasingly relevant. Privacy can be defined as the demands from individuals, groups and institutions to determine by themselves when, how and to what extent information about them is to be communicated to others [3GPPa].

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 121–135, 2010.

We define privacy as the right of individual to control their personal data. By personal data we mean any information that can be used to identify directly or indirectly a person, who is the data subject or the owner of the information.

Currently, enhanced services extensively use sensitive information. We must first stress the fact that data controller, which collects sensitive information, is different from the service provider, which uses these information to offer services.

In this paper, we focus on managing the sensitive information when they are collected by a data controller. We will formalize the privacy policy that a data controller should implement to protect the privacy of their subscribers.

'Protection of Privacy and Transborder Flows of Personal Data' was a major guideline that defined the cornerstones of privacy requirements in various countries [Audy 06]. The guidelines continue to represent international consensus on general guidance concerning the collection and management of personal information. It was instituted by the Organisation for Economic Co-Operation and development (OECD) in 1980. Eight privacy principles were defined: collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation, and accountability.

Most dedicated models, which enforce privacy, are mainly based on access control models because they are interested in the protection of sensitive information. Enhanced privacy-aware models are proposed, such as P-RBAC [Ni et al 2007], Purpose-BAC [Yang et al 2008] and Pu-RBAC [Masoumzadeh and Joshi 2008], to take into account purposes and obligations.

In this paper, we propose to use the OrBAC model [Cuppens 2007] to express privacy policies. We specify the most relevant privacy requirements, which are the consent, accuracy, provisional obligations and purposes. Our proposal extends the existing model with a new consent context. On the other hand, we model the object's accuracy and the provisional obligations that requesters and providers must perform after accessing sensitive data. We show that dynamic contexts could be expressive enough to take into account the major privacy principles.

The paper is organised as follows. Section 2 lists the privacy requirements that will be deployed through access control models. Section 3 is dedicated to the privacy-aware OrBAC model. We will present the OrBAC model and propose the enforcement of consent context, purpose and provisional obligations. Section 4 presents a use case of location service and how a privacy policy is specified through the OrBAC model. Related works are presented in section 5. Concluding remarks are presented in section 6.

2 Privacy Requirements and Motivation

We illustrate in this section the issues related to private data management and how to use a privacy policy to specify privacy requirements. Private data are collected by mobile operator networks since we focus, in our work, on sensitive data such as location and presence of mobile subscribers that only the operator

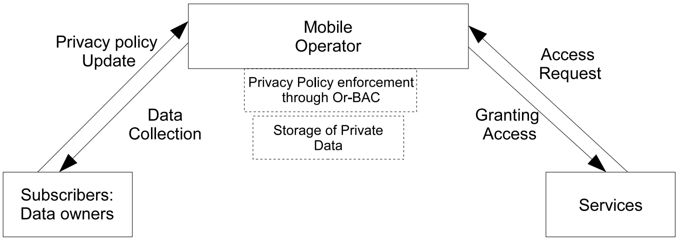


Fig. 1. Use case of privacy enforcement

network can collect them. At this stage we do not care about means used to collect data. Collected data concerns operator's subscribers.

The information is stored within operator's information system. The later should implement the OrBAC model to enforce the privacy policy defined by the subscribers. Service providers request that information to offer enhanced services. So, the operator should manage the access to services.

2.1 Privacy Requirements

According to the privacy principles, we can identify the following goals of the privacy policy:

- The definition of purpose: before delivering private data, third parties should mention the purpose of the access request. This requirement corresponds to the purpose specification principle, which is instituted by OECD,
- The definition of the accuracy of the private data: this parameter is a data owner one. The user can set its preferences by choosing the accuracy, which corresponds to the level of anonymity and/or to the level of the remaining data. This parameter enforces the principles of collection limitation and use limitation,
- The user consent: data owner preference can include the requirement of consent before delivering the personal data to third parties. The notification is on the fly, so the access is pending until the consent of the data owner. That requirement is exclusively mentioned in the use limitation principle,
- Obligations after the access: the data controller, which collects personal data has to ensure the usage of personal data. This corresponds to some obligations ordered to third parties. Usage control is a relevant concept to enforce privacy requirements. It is mentioned by the principles of use limitation and accountability.

2.2 Motivation for Using the OrBAC Model to Express Privacy Requirements

Integrating privacy requirements into a security policy, expressed through an access control model will permit an ease upgrade of existing information systems, which already implemented access control policies.

The OrBAC model presents some interesting characteristics:

- The support of environment and dynamic parameters through contexts,
- The definition of a security policy within the organization. That corresponds to the fact that privacy practices usually apply to an organization on which data owners define their preferences. Users also specify obligations that requesters must perform after access sensitive information.

For this, we chose the OrBAC model to include privacy requirements. We now extend the OrBAC model by proposing a new consent context and a view hierarchy to take into account object's accuracy.

3 Privacy-Aware OrBAC

3.1 OrBAC

Organization-Based Access Control model (OrBAC) [Cuppens 2007] is an innovative access control model. It provides interesting mechanisms to express the security policy and enables making distinction between an abstract policy specifying organizational requirements and its implementation in a given information system.

Traditional models are based on subjects that have right to make actions on objects. An abstraction level is offered by OrBAC to abstract subjects into role, objects into views and actions into activity. This abstract level is introduced to design implementation-independent policies. Those entities are designed within an organization to permit interoperability between organizations and enforce separation of duties. The organization is the central component. It groups a set of subjects and it is in charge of defining and enforcing the security policy applied to subjects. So, the specification of the policy is parameterized by the organization. This is used to handle simultaneously several security policies associated with different organizations. Privileges do not directly apply to subjects, they are assigned to roles within an organization [Cuppens 2007].

Abstract organization privileges, such as permission, are expressed through the predicate:

Permission (organization, role, activity, view, context).

OrBAC authorizes the use of four kinds of privileges: permission, prohibition, obligation and dispensation. They mean that a given role, is permitted, respectively prohibited, obliged or dispensed, to perform a given activity on a given view.

A privilege corresponds to a relation between roles, views and activities at the organizational level. The concrete policy is logically derived from abstract privileges, according to derivation rules, when request for an access is received by the policy controller. The correspondent derived concrete privileges are Is_permitted, Is_prohibited, Is_obliged and Is_dispensed. They compute if a given subject, belonging to a role, can perform a given action, belonging to an activity, on a given object, belonging to a view.

In OrBAC, there are three built-in predicates, which specify conditions over subjects, actions and objects:

- *Empower* is a predicate over domains *Org* $\times S \times$ *R*. If *org* is an organisation, s a subject and r a role, then *Empower(org, s, r)* means that s play the role r within *org*,
- *Consider* is a predicate over domains *Org* $\times A \times$ A. If org is an organisation, α an action and a an activity, then *Consider,* α*, a* means that org considers that α is implementing the activity a,
- *Use* is a predicate over domains *Org* $\times O \times$ *V*. If *org* is an organization, o is an object and v is a view, then *Use(org, o, v)* means that org uses the object o in the view v.

Contexts are designed to take into account dynamic parameters of a security policy. A context is defined as an abstract condition that takes into account such environment parameters when specifying abstract organization privileges. So, contexts are designed to allow the definition of a dynamic security policy. Contexts are constraints that model extra conditions a subject, an action and an object must satisfy to activate a privilege. An OrBAC built-in predicate *Hold* permits linking those entities:

- *Hold* is a predicate over domains *Org* $\times S \times A \times O \times C$. If *org* is an organization, s is a subject, α is an action, o is an object and c is a context, then *Hold(org, s,* α*,o, c)* means that context c holds between subject s, action α and object o within *org*.

The OrBAC model defines five types of contexts [Cuppens 2007]:

- Spatial context: that depends on the subject position,
- Temporal context: that depends on the time of the subject request,
- User-declared context: that depends on parameters declared by the subject,
- Prerequisite context: that depends on a relation between the subject, the action and the object,
- Provisional context: that depends on the previous actions of the subject.

To structure the set of entities and authorizations in OrBAC, hierarchies and inheritance mechanisms are introduced. Both roles, Views, Activities, Contexts and Organization may be structured according to hierarchies. In this case, privileges are inherited through this hierarchy. For this purpose, we define the predicate sub_role to specify role hierarchies and similar other predicates for activity, view, context and organization hierarchies.

3.2 Modelling Privacy within OrBAC

In order to add privacy requirements listed in previous section, we propose to model (figure 2):

- The consent context,
- Object's hierarchy based on the accuracy of objects,
- The purpose as a user-declared context,
- Provisional obligation following the access to sensitive information.

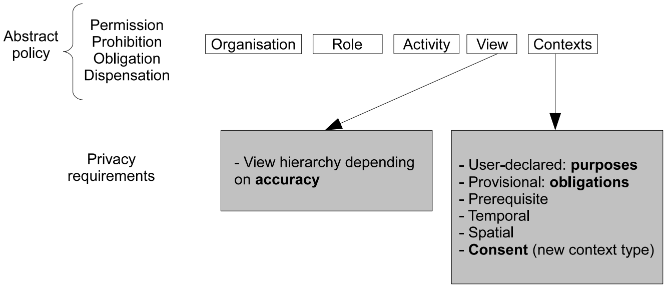


Fig. 2. OrBAC model

3.2.1 Consent Context

3.2.1.1 Principle. The consent context is a relevant parameter in the privacy preference. It permits the notification of the data owner when his personal information is accessed.

Furthermore, 3GPP recommends using consent for location service in two ways:

- Before data collection,
- After data collection.

By this way, the user can control when it is localized and if the position, after being computed, hinders his privacy or not.

We are interested in the second case which corresponds to the case where data controller collects yet sensitive data.

For instance, data owner can require that subjects, acting in some given role, must give their consent before granting the access. The permission privilege can enter a pending mode waiting for the consent. For all other roles, the access will be denied. We consider that the data owner is the subscriber that the sensitive information or the collected object is referred to. For each collected object, a 'data owner identity' attribute is associated with the object. The consent context evaluates if the data owner preference requires a consent or not.

3.2.1.2 Consent principle. When the operator receives an access request from the service provider, we suppose that the operator should maintain a trace about the data owner's preference regarding the need of consent or not. This need depends on the data owner. The trace (or the history) of users preference is modelled by a view *consent preference.*

Each object of the later view has four attributes:

- *Requestor*: the subject who requests the access to the object,
- *Target*: the requested object,
- *Data-owner*: the subscriber that the location or the object referred to,
- *NeedConsent*: it is a Boolean parameter. If it is true, that means a consent must be given by the data owner before granting the access.

Now, we assume that the subscriber's consent (after being notified by the operator) is associated with a built-in predicate *Consent_ response*. It is a predicate over domains *Org* × *Data_ owner* × *Subject*.

We are now ready to specify when user consent is triggered:
$\forall org \in organisation, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall cp \in O,$
$Hold(org, s, \alpha, o, Consent\_context) \leftarrow Use(org, cp, Consent\_preference)$
$\wedge\, Requestor(cp, s) \wedge Target(cp, o) \wedge Data\_owner(cp, do) \wedge NeedConsent(cp)$
$\wedge\, Consent\_response(Org, do, s)$
The above formula means that the *Consent_ context* is triggered only if:

- there is an object *cp*, belonging to the *Consent_ preference* view, that has the attributes s as the requestor, o as the target, do as the data owner and the *NeedConsent* attribute,
- and the *Consent_ response* predicate is true.

3.2.1.3 Example of consent context. Let 'mobile_subscriber' be a mobile subscriber that defines the use of consent before granting access to the location information. And, let be fm a fleet management service provider.

In this case, we have an object belonging to the *consent preference* view with the following attributes:

- *Requestor*: fm,
- *Target*: location information,
- *Data_ owner*: mobile_subscriber,
- *NeedConsent*: true.

$\forall org \in organisation, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall cp \in O,$
$Hold(org, fm, \alpha, location\_information, Consent\_context) \leftarrow$
$Consent\_response(Org, mobile\_subscriber, fm)$

3.2.2 Accuracy Attribute

Privacy enforcement requires the use of different levels of accuracy depending on the purpose and the subject requesting the access to the private data. That principle is consistent with the privacy directive of collection limitation since service providers cannot access more accurate objects than user preference and needed accuracy for the service. First, we define the view 'location data' that groups all location information of subscribers.

Then, we suggest that private objects have different levels of accuracy. We propose to institute a hierarchy between the root view, which groups the initially collected objects, and sub-views, which group the derived objects that have different accuracies.

3.2.2.1 Accuracy levels. Before defining the hierarchy of location data, we will propose an accuracy model for private data. We will focus on the special case of the location data.

Each location object has two attributes: the identity of the data owner and the location itself. Each attribute has different levels of accuracy.

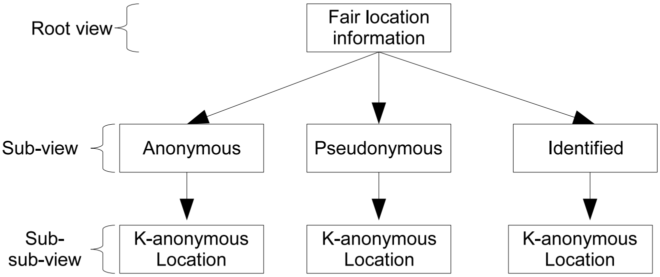


Fig. 3. Accuracy levels of location data

We distinguish between those two attributes because they are loosely coupled. Malicious third parties can hinder user privacy only if they can break the relation that binds the accurate position with the identity of the subscriber.

Figure 3 shows that accuracy is specified by the couple (anonymity level, k):

- Identity is an attribute of the location information. The 'anonymity level' defines how the subscriber identity is masked. There are three levels:
  - Anonymize user: no identity is used,
  - Pseudonymity: a temporal pseudonym is used to identify the location information,
  - Fair identity: the subscriber chooses to use his identity (name or phone number) to identify the location information.
- k: this parameter is used in the chosen k-anonymity algorithm. The subscriber is indistinguishable within some zone area between k other subscribers. There is a plethora of such algorithms [Gedik and Liu]. However, for other sensitive information similar algorithms can be applied to render a user attribute indistinguishable within a set of k other attributes.

3.2.2.2 Accuracy-based object hierarchy. Figure 3 shows two stages in the hierarchy of the location views:

- The first one is computed by replacing the identity attribute by a pseudonymous, anonymous or the true identity itself,
- The second stage is computed by executing the k-anonymity algorithm over location attributes (coordinates, vicinity). The implemented algorithm depends on the operator choice.

Subscribers will define chosen processing on their location information by defining the k parameter. Then, subscribers can set their privacy policy over those resulting objects. 'Permission privilege' is granted to the authorized service providers by subscribers. This administration task will be investigated in further work. And for this purpose, we plan to use AdOrBAC, the administration model associated with the OrBAC model [Cuppens and Miège 2004].

3.2.2.3 Object derivation rules. Objects are derived from a root object based on the accuracy. Let RO be the root object belonging to the root view RV.

The data owner of sensitive data defines a set of accuracies through its privacy preferences.

We define accuracy as:

$$anonymity\_level = \{fair, anonymous, pseudonymous\}$$
$$accuracy : anonymity\_level \times N$$

We suppose that the data controller implements an anonymity algorithm $algo_k$, which deprecates the position of a user until a zone area including k users. k is an entry parameter set by the data owner.

$$\forall ro \in RO, \forall accuracy(a, b), a \in anonymity\_level, b \in N$$

Let identity and position be the two attributes of ro defining the identity of the data owner and his position respectively. The derived object dr is computed as follows:

$$\begin{array}{ll}
a = fair & \rightarrow identity(dr) = identity(ro) \\
a = pseudonymous & \rightarrow identity(dr) = pseudonym \\
a = anonymous & \rightarrow identity(dr) = \emptyset \\
b > 0 & \rightarrow position(dr) = algo_b(position(ro)) \\
b = 0 & \rightarrow position(dr) = position(ro)
\end{array}$$

Now, we define the views that group the derived objects *dr*. Let anonymous, pseudonymous and identified be three views within mobile-operator organization:

$$\begin{array}{ll}
identity(dr) = identity(ro) & \rightarrow Use(mobile\_operator, dr, identified) \\
identity(dr) = pseudonym & \rightarrow Use(mobile\_operator, dr, pseudonymous) \\
identity(dr) = \emptyset & \rightarrow Use(mobile\_operator, dr, anonymous)
\end{array}$$

3.2.3 Purpose as a User-Declared Context

We suggest modelling purpose as a user-declared context. In OrBAC model, the definition of a user-declared context consists in two steps:

1. Specification of roles who can declare this user-declared context,
2. Specification of roles that are permitted to fulfil a given activity when they declare the associated user-declared context.

We assume that purposes are grouped into a *Purpose* view. Purpose objects are a finite set denoted *PO*. Each purpose object has two attributes [Cuppens 2007]:

- *Recipient*: who takes advantage of the declared purpose (the service provider in our case),
- *Declared_purpose*: is a predicate to associate the purpose value with the declared purpose.

We aim to use this OrBAC capability to enhance privacy enforcement. First, Subscribers, who play the ‘subscriber’ role, defines which purpose to be defined by service providers before accessing location information. So, ‘subscriber’ is the role that declares the context. The purpose that we want to model is ‘optimise_route’.

Permission(mobile_ network, subscriber, declare, optimise_ route, subscription)

Where the subscription context is defined as follows:

$\forall s \in subscriber, \forall s' \in service\_provider, \forall \alpha \in A, \forall po \in PO$
$Hold(mobile\_network, s, \alpha, po, Subscription) \quad \leftarrow \quad Use(mobile\_network, po, Purpose)$
$\wedge\ Recipient(po, s') \wedge Existing\_subscription(s, s')$

This rule says that a subject s is in context ‘Subscription’ if there is a purpose object po having s’ as recipient and where s and s’ have an agreement modelled by the application dependent predicate Existing_subscription(s, s’).

The second step defines roles that take advantage of this context. In our case the ‘service provider’ is the recipient that takes advantage of the context. We specify that service providers can access the location information view if they declare a purpose context:

Permission(mobile_ network, service_ provider, consult,
location_ information, user_ declared(optimise_ route))

3.2.4 Requirements Beyond Granting the Access: Provisional Obligation

Since private data can be stored and reused for unauthorized purposes, we define usage control over private data. Basically, usage control is introduced thanks to obligation requirements.

Obligations are yet introduced within the OrBAC model [Cuppens 2007]. Thanks to ‘provisional obligation’, the operator can order service providers to perform some action following its access to the location information. The obligation is automatically triggered as a counterpart of the access to the private information.

As defined in [Cuppens 2007], the history of access requests should be stored in the *log* view. Objects belonging to the *log* view have five attributes: *Log_ actor, Log_ action, Log_ target, Log_ date and Log_ context.* Those attributes represent the subject who performs an action on an object at a given date and in a given context. A reference monitor is responsible for inserting entries into *log* objects.

The obligation is expressed thanks to two types of contexts *Context_ activation* and *Context_ violation*:
Obligation(org, role, activity, view, Context_ activation, Context_ violation)
This privilege means that the obligation has to be performed once *Context_ activation* becomes true, and will be violated if *Context_ violation* before the activity is fulfilled.

For example, within mobile network organization service provider has to send details about provided service, *Context_ activation* will be ‘providing location service’, while *Context_ violation* could be a time period of one day if details must be send in the same day.

3.2.5 Provisional Obligation as Privacy Requirement for Location Service

The service provider cannot ignore the obligation because the existing agreement between service provider and the operator. Furthermore, the operator can order that external legal entity will supervise the usage of the sensitive information.

A service provider has to notify subscribers about further usage, such as data deletion. We assume that the notification obligation is delivered to the operator that will handle it to the subscriber. We are concerned only about the notification to the operator. The interaction between the operator and the subscriber is out of the scope of this work.

1. For this, we define a provisional context 'Notification' within the operator organisation:

$$\forall s \in Serviceprovider, \forall \alpha \in A, \forall o \in O, \forall l \in O$$
$$Hold(operator, s, \alpha, o, Notification) \leftarrow Use(operator, l, log) \wedge Log\_actor(l, s) \wedge$$
$$Log\_action(l, \alpha) \wedge Log\_target(l, o) \wedge Log\_context(l, Notification)$$

Where l is an entry into the log file.

2. We then define a view 'Notification_list' that groups the notifications received by the operator from service providers. Objects in this view have three attributes: service provider identity, content of the notification and subscriber to which the notification can be handled later. The context of the notification can be for example 'data deleted' or the purpose of further use.

3. Third, we define the obligation that stipulates: a service provider should notify operator for further use on private data within a time period of 30 seconds: *Obligation(operator, serviceProvider, notify, Notification_ list, Notification, 30)*

4 Use Case

4.1 User Preference Scenario

Based on the privacy policy for Location Service (LCS), we propose to study the sample of a the privacy policy instituted by a mobile subscriber to protect his location data.

The subscriber will define which service providers can access his location information. He grants the access to the providers that offer a 'fleet management' service in order to optimise the route undertaken by him. (The service can be offered to a group of users to optimise their movements.)

The user is scared to be localized at holidays when he is visiting some points of interest. The subscriber aims that the positioning can be performed only when it is done in working hours and within a predefined area.

Then, user distinguishes between two service providers: 'fleet management 1' and 'fleet management 2'. The two service providers are associated with two accuracies. 'fleet management 1' can access to the location with an 'accuracy 1' and 'fleet management 2' can access to the location with an 'accuracy 2'. 'Accuracy 1' is higher than 'accuracy 2'. When 'fleet management 1' requests access, the subscriber must give his consent before granting access.

When ‘fleet management 2’ access the private data, a provisional obligation is defined to be fulfilled after the access. The service provider has to inform the operator when the location information is deleted.

4.2 Privacy Requirements

The privacy requirements can be summarized as follows. ‘fleet management 1’ and ‘fleet management 2’ have the role of ‘fleet management service provider’. Those service providers must have contracted an agreement with the subscriber to offer the service. So, a subscription must exist before requesting access to private data. The first subject, which is the ‘fleet management 1’, can access the location object if it is anonymized however the second subject can access the object if it is pseudonymized.

The purpose of the access is ‘optimise route’. The temporal context institutes that the access is granted only it is done in ‘working hours’. Furthermore, a consent is needed before delivering location data for ‘fleet management 1’.

An obligation has to be fulfilled by ‘fleet management 2’ when location data is accessed. The obligation specifies that location data deletion must be notified.

4.3 Specifying the Privacy Policy of the Use Case through the Privacy-Aware OrBAC Model

Now, we illustrate how our use case can be specified through the OrBAC model.

The policy is defined within the operator (mobile network) organisation to control the access of the service provider. Let ‘trade_representative’ be a subscriber in the organisation ‘mobile_network’, and ‘oil_company’ a service provider. The subscriber is considered as a subject within the organisation. We define:

- The role ‘subscriber’
  Empower(mobile_network, ‘trade_representative’, ‘subscriber’)
- The role ‘service_provider’
  Empower(mobile_network, ‘oil_company’, ‘service_provider’)
- Two sub-roles: ‘fleet_management_1’ and ‘fleet_management_2’
  Sub-role(mobile_network, fleet_management_1, service_provider)
  Sub-role(mobile_network, fleet_management_2, service_provider)
- The activity ‘Consult’
- The view ‘Location_view’
- The view hierarchy depending on the accuracies: ‘pseudonymous_view’ and ‘anonymous_view’

  Sub-view(mobile_network, pseudonymous_view, Location_view)
  Sub-view(mobile_network, anonymous_view, Location_view)
  pseudonymous_view and *anonymous_view* are defined analogically than *‘pseudonymous’* and *‘anonymous’* views in 3.2.2.3,
- The purpose ‘optimise_route’ as a user-declared context

- The temporal context is ‘working_hours’
- The ‘consent_context’ means that the permission is granted only if the data owner is notified and gives its consent (section 3.2.1.3)
- The provisional context ‘Notification’ that defines provisional obligations (section 3.2.5). The notification must be done within 30 seconds.

We define the permissions that enforce privacy policy within the operator organisation for our use case. The first permission allows ‘fleet_management_1’ to access anonymous ‘location_information’ for ‘optimise_route’ during ‘working hours’ and if the subscriber gives his consent:

- *Permission(mobile_network, fleet_management_1, consult, anonymous_view, optimise_route & working_hours & consent_context)*

The second permission allows ‘fleet_management_2’ to access pseudonymous ‘location_information’ for ‘optimise_route’ during ‘working hours’ and if the service provider notifies the subscriber about the data deletion:

- *Permission(mobile_network, service_provider, consult, pseudonymous_view, optimise_route & working_hours & Notification)*

Since ‘Notification’ context is specified, the following obligation will be triggered:

Obligation(operator, serviceProvider, notify, Notification_list, Notification, 30s)

4.4 Conformity with Privacy Requirements

The proposed policy specification tends to enforce major privacy principles as they are instituted by OECD. We illustrate how those principles are implemented in our policy.

- Purpose specification principle is expressed through a user-declared context. The service provider has to mention the objective of its access before processing the access request,
- Collection limitation principle is naturally defined through the use of different accuracies that limit the access of service providers,
- Data quality principle stipulates that collected data must be accurate but we proposed to let the choice to subscribers to define a view hierarchy depending on the accuracy of objects,
- Accountability and use limitation principles are enforced by provisional obligations. From the service provider point of view, it is responsible for notifying operator about further use applied over the private information,
- Individual participation principle means that individual can access they own data to modify, rectify and suppress it. It is easy to model that through new privileges where data owner can execute those actions in the context of ‘personal_information’. We omitted those privileges due to space limitation,
- Openness and security safeguards principles are omitted in our policy specification because they are implementation-dependent.

5 Related Works: Access Control Models and Privacy

Existing access control models do not consider privacy protection as their premier goal. Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role-Based Access Control (RBAC) meet few privacy requirements [Ni et al 2007]. Qui Ni et al., in [Ni et al 2007], proposed RBAC extensions to incorporate constraints and conditions that are related to privacy. They define a family of privacy aware RBAC (P-RBAC) models. A user is a human being when a role is a job title or a job function. A customized language, LC_0, allows the definition of conditions. A privacy permission explicitly defines: the intended purposes of the action, under which conditions, and what obligations have to be performed after the access.

So, the two main extensions are: obligation definition and a dedicated language for conditions. In [Ni et al 2007] privacy policy is enforced by permission assignments. The Purpose-Based Access Control model was proposed in [Yang et al 2008]. It is more concerned on the formalization of purposes and obligations. It provides proofs of privacy invariants. Authors aim to enforce privacy in non-trusted domains.

Purposes are divided into two classes: intended purposes class, which groups the intended usage of data element, and access purposes class, which groups the intentions for which data are accessed. Access purpose should be compliant with intended purpose to authorise the access.

The definition of role entity in RBAC was extended to include conditional role, which is based on role attributes and system attributes [Byun et al 2005]. Also, a key characteristic of this work is that several purposes may be related to each data element. This model was deployed in relational databases.

In our proposal, our concept of role is different from conditional role. We argue that a role cannot be modified by some attributes after being assigned by the administrator.

A Purpose-Aware Role-Based Access Control model (PuRBAC) is proposed in [Masoumzadeh and Joshi 2008]. It extends RBAC by modelling privacy requirements. Purposes are the central entity, they are the intermediary entity between role and permission entities.

The model defines constraints and obligations as conditions on assignment of permissions to purposes. Then purposes are assigned to roles.

In this model, users are assigned to roles, purposes are assigned to role, permissions are assigned to purposes, and conditions are assigned to permission. A user requests is formed by a session, purpose and requested permission. Authorisation can be requested only for purposes related to the active role.

There is another major difference with the RBAC model. When a user request is submitted to the Access Decision Function (ADF), it either denies access or defines a conditional authorization. Authors model three types of conditions:

- Constraints: they include are used to check information based on data variables in the system. For instance, the consent of the data owner is considered as a constraint,

- Pre-obligations: the system or the user has to exercise some actions before granting the access. They include for example: the re-authentication of the user before accessing sensitive data, or the readjustment of the data accuracy,
- Post-obligation: they include for example a data retention policy that would schedule data deletion.

Cited works share our objective to model privacy within the access control policy since both policies manage access to the same resource. Those models are based only on purposes. We argue that purpose is not sufficient for users to define their privacy preferences. We present limited changes in the OrBAC model. We defined a new context type: *Consent* context. Furthermore, we showed that purposes and provisional obligations are expressed thanks to existing context types: user-declared and provisional contexts respectively. The accuracy is introduced by defining a view hierarchy of sensitive objects based on user preferences. Those concepts are sufficient to be conform to privacy principles and they take into account more privacy requirements than listed models.

6 Conclusion

We showed in this paper how privacy preferences could be integrated in an access control policy. We enforced the privacy policy thanks to the OrBAC model augmented with a consent context to deal with some privacy requirements.

The OrBAC model is expressive enough to handle dynamic parameters thanks to contexts in order to implement user privacy preferences.

We plan to investigate how subscriber can set their privacy settings. Privacy settings will impact the OrBAC policy enforced by the operator.

References

[3GPPa] 3rd Generation Partnership Project: Open Service Access; Application Programming Interface (API); Part 3: Framework, 3GPP TS 29.198-3

[Audy 06] Audy, S.: Le respect de la vie privée et la protection de la confidentialitéen recherche. Comité de liaison en éthique de la recherche de l'Université de Montréal (CLERUM), Canada, Mars (2006)

[Byun et al 2005] Byun, J., Bertino, E., Li, N.: Purpose Based Access Control for Complex Data for Privacy Protection. In: SACMAT, Stockholm, Sweden (2005)

[Cuppens 2007] Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual Security Policies. International Journal of Information Security (2007)

[Cuppens and Miège 2004] Cuppens, F., Miège, A.: An Administration Model for Or-BAC. International Journal of Computer Systems Science and Engineering (May 2004)

[Gedik and Liu] Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing (2007)

[Masoumzadeh and Joshi 2008] Masoumzadeh, A., Joshi, J.B.D.: PuRBAC: Purpose-Aware Role-Based Access Control. In: OTM, Mexico (2008)

[Ni et al 2007] Qui Ni, A., Trombetta, E., Bertino, J.: Privacy-aware Role Based Access Control. In: 12th ACM symposium on Access control models and technologies (2007)

[Yang et al 2008] Yang, N., Barringer, H., Zhang, N.: A Purpose-Based Access Control Model. Journal of Information Assurance and Security (2008)

Dynamic Security Rules for Geo Data*

Alban Gabillon and Patrick Capolsini

Université de la Polynésie Française, Laboratoire GePaSud,
BP 6570 – 98702 FAA'A – Tahiti – Polynésie française
{alban.gabillon,patrick.capolsini}@upf.pf

Abstract. A powerful and flexible authorization model should be able to cope with various security requirements. We show in this paper that we can use the Or-BAC model [1] to express security policies for spatial applications. We first add to Or-BAC the spatial predicates defined in the OpenGIS Geometry Model [2]. We then show how to model various types of spatial contexts. We finally use these spatial contexts to write security policies for spatial applications.

Keywords: Security Policy, Spatial Context, Geo-Referenced Object, Moving Object.

1 Introduction

The core RBAC [3] authorization model considers only static security rules. However, in many applications, there is an increasing need for dynamic security rules. A dynamic security rule can be activated/deactivated depending on some *context*. A context can be a temporal condition, a spatial condition (like the user location), a provisional condition (like the user previous action) etc. Therefore, several extensions to the RBAC model have been proposed in order to cope with contexts: the Generalized Role Based Access Control (GRBAC) [4] incorporates the notion of object role and environment role; in the Context-Role Based Access Control (CRBAC) [5] some constraints should be fulfilled before a permission is assigned to a role; the Or-BAC model [1] allows the security policy designer to express various types of contexts, by using first-order logic. Some models focus on specific contexts, like temporal contexts: the Temporal Role Based Access Control Model (TRBAC) [6] offers means to activate roles periodically; the Generalized TRBAC (GTRBAC) [7] incorporates various temporal constraints on role activation as well as on user-to-role or permission-to-role assignment.

With the growing importance of geographic information in various applications, there is a need for dynamic security rules based on some *spatial contexts*. Therefore, security models specifically dealing with spatial contexts have also started to appear: the GeoRBAC [8] model introduces the concept of spatial role to specify spatial condition on user location; the GSAM [9] model introduces the extended concept of geo-temporal role. There are in fact different types of spatial contexts. A spatial context

* This work was conducted as part of the ANR funded project under reference ANR-SESUR-2007-FLUOR.

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 136–152, 2010.

can be the position of a user (in a Location Based System), the zoom level at which a user is looking at a map, the direction followed by a moving object etc. In this paper our objective is twofold: it is first to identify and model various types of spatial contexts and then to model security rules based on such contexts. For achieving our goal, we have the following three possibilities:

1. We can use an existing security model for spatial applications like GeoRBAC.
2. We can use an existing generic authorization model like Or-BAC.
3. We can define our own security model from scratch (clearly we should opt for this solution only if we fail at representing certain types of spatial contexts with existing models).

We believe that existing models for spatial applications like GeoRBAC and GSAM are not flexible enough to be adapted to all kinds of spatial applications. Neither GeoRBAC nor GSAM includes the concept of spatial context. In order to express spatial conditions in the security policy, they rather use the non intuitive concept of *spatial role*. However, spatial roles do not allow to express all kinds of spatial conditions. Moreover, spatial roles make the security policy management more complex.

To our knowledge, the Or-BAC model is the only authorization model which allows the security policy designer to express various types of contexts within a single framework. Or-BAC is a generic security model which formally defines the notion of context and offers a language based on first order logic to specify them. In [10] and [11], the authors show how to define various types of contexts (including temporal contexts and provisional contexts) with this language. Therefore, we have decided to use Or-BAC for writing security policies for spatial applications. Having said this, we can now reformulate our objective as follows:

1. We add to Or-BAC some spatial functions and predicates for expressing spatial conditions
2. We try to figure out various types of spatial contexts
3. We show how to model these contexts with Or-BAC
4. We show how to write security rules based on spatial contexts

The remainder of this paper is organized as follows. Section 2 recalls the basic principles of the Or-BAC model. Section 3 defines our geometry model which is based on the OpenGIS [12] geometry model [2]. In section 4, we identify several types of spatial contexts and we show how to model them with Or-BAC. In section 5, we describe a spatial application and we give an example of security policy based on spatial contexts. In section 6, we compare our work with other authorisation models for geospatial data. In particular we show that some spatial contexts cannot be expressed neither with GeoRBAC nor with GSAM. Finally, in section 7, we conclude this paper and suggest some future extensions.

2 Or-BAC

In Or-BAC [1], there are eight basic sets of entities: *Org* (a set of organizations), *S* (a set of subjects), *A* (a set of actions), *O* (a set of objects), *R* (a set of roles), *T* (a set of activities), *V* (a set of views) and *C* (a set of contexts). $Org \subseteq S$ (any organization is also a subject) and $S \subseteq O$ (any subject is also an object). Subjects, actions and objects are respectively abstracted into *roles*, *activities* and *views*. Roles, activities and views are the *abstract entities* and are always created within the framework of an organization. Abstract entities are organised into hierarchies [13]. Subjects, actions and objects are the *concrete entities*. Each subject (resp. action and object) is linked to one or several roles (resp. activities and views). Abstract entities and concrete entities are linked together by the relations *Empower*, *Use* and *Consider*. *Empower* is a relation over domains $Org \times S \times R$. If *org* is an organization, *s* a subject and *r* a role, then *Empower(org,s,r)* means that organization *org* empowers subject *s* in role *r*. *Use* is a relation over domains $Org \times O \times V$. If *org* is an organization, *o* an object and *v* a view, then *Use(org,o,v)* means that organization *org* uses object *o* in view *v*. *Consider* is a relation over domains $Org \times A \times T$. If *org* is an organization, *a* an action and *t* an activity, then *Consider(org,a,t)* means that *org* considers that action *a* falls within the activity *t*. Any entity in the Or-BAC model may have some *attributes*. This is represented by functions that associate the entity with the value of these attributes. For instance, if *s* is a subject, then *name(s)* represents the value of attribute *name* of *s*.

A context is ***any kind of constraint*** which may or may not involve the subject and/or the action and/or the object. Organization, subject, object, action and context are linked together by the relation *Hold*. *Hold* is a relation over domains $Org \times S \times A \times O \times C$. If *org* is an organization, *s* a subject, *a* an action, *o* an object and *c* a context, then *Hold(org,s,a,o,c)* means that within organization *org*, context *c* holds between subject *s*, action *a* and object *o*. For example, a context *Teacher* can be defined as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(University\_FrenchPolynesia, s, a, o, Teacher)$$
$$\leftrightarrow name(o) \in students(s)$$

that is, at organization University of French Polynesia, context *Teacher* holds between subject *s*, action *a* and object *o* if and only if object *o* is a record corresponding to a student of subject *s*.

There is one default context *Default_ctx* which is always true. In [11], the authors show how to represent different types of contexts with Or-BAC, namely temporal context, user-declared context, prerequisite context and provisional context. They also define some simple spatial contexts using a single built-in predicate *Is_located*.

In Or-BAC, the *security policy* is specified using the relationships *Permission*, *Obligation* and *Prohibition*. *Permission*, *Obligation* and *Prohibition* are relations over domains $Org \times R \times T \times V \times C$. If *org* is an organization, *r* a role, *t* an activity, *v* a view and *c* a context then *Permission(org,r,t,v,c)* (resp. *Obligation(org,r,t,v,c)* or *Prohibition(org,r,t,v,c)*) means that in organization *org* role *r* is granted permission (resp. obligation or prohibition) to perform activity *t* on view *v* within context *c*. Instances of *Permission*, *Obligation* and *Prohibition* are called *abstract rules*. These abstract rules are propagated downwards in hierarchies of roles, activities and views through an inheritance mechanism (see [13]). Note that in this paper, for the sake of simplicity, we shall not consider obligations. Indeed, our paper focuses more on how to model spatial contexts than on how to express security rules. For modelling security rules, we apply Or-BAC principles as such. The reader who is interested can refer to [11] where obligations and related concepts of user declared context and provisional context are described in detail. *Concrete rules* are instances of the relationships *Is_permitted*, *Is_prohibited* and *Is_obliged. Is_permitted*, *Is_prohibited* and *Is_obliged* are relations over domains $S \times A \times O$. Instances of these relationships are logically derived from the abstract rules. The following rule allows us to derive instances of *Is_permitted* from the relation *Permission*:

$$\begin{aligned}&\forall org \in Org, \forall s \in S, \forall o \in O, \forall a \in A, \forall r \in R, \forall v \in V, \forall t \in T, \forall c \in C,\\ &Permission(org, r, t, v, c) \wedge Empower(org, s, r) \wedge Use(org, o, v) \wedge\\ &Consider(org, a, t) \wedge Hold(org, s, a, o, c) \rightarrow Is\_permitted(s, a, o)\end{aligned}$$

that is, if organization *org*, within context *c*, grants role *r* permission to perform activity *t* on view *v* and if *org* empowers subject *s* in role *r* and if *org* uses object *o* in view *v* and if *org* considers that action *a* falls within the activity *t* and if, within *org*, context *c* holds between *s*, *a* and *o* then *s* is permitted to perform *a* on *o*. There is a similar rule for *Is_prohibited* and *Is_obliged.* Specifying a security policy that includes both permissions and prohibitions may lead to conflicts. The Or-BAC model makes the distinction between the *potential conflicts* between abstract rules and the *actual conflicts* between instances of the *Is_permitted* and *Is_prohibited* predicates. The conflict resolution strategy in Or-BAC acts at the abstract level and is based on two complementary approaches : *separation constraints* and *rules priorities*, leading to the concept of prioritized Or-BAC [14]. Since a subject can potentially be empowered in different roles, an object can be used in different views, an action can fall within different activities and different contexts can be active simultaneously, every pair of Permission and Prohibition may be potentially conflicting. Such potential conflicts can be eliminated by specifying separation constraints. For instance, if a separation constraint exists between roles r_1 and r_2, then no subject can be empowered in both roles and a Permission assigned to role r_1 cannot get into conflict with a Prohibition assigned to role r_2. Remaining conflicts are solved by explicitly assigning priorities to abstract rules.

3 Geometry Model

The Or-BAC language for specifying contexts is based on first-order logic. We extend this language with spatial attributes and spatial methods, some geometric functions and some spatial predicates (refer to [10] for other aspects of this language).

3.1 Geometric Objects

A *georeferenced (geometric) object* is a granule of information that is relevant to an identifiable subset of the Earth's surface [15]. Any geometric object has the following two components [16] : a *description*: the entity is described by a set of descriptive attributes (e.g. the name of a city) and a *geometry* which indicates the entity’s location and its shape.

The geometry model we consider is the OpenGIS Geometry Model [2]. In this model, each geometric object belongs to a geometry class. In this paper, we do not consider the whole class hierarchy defined in [2]. For the sake of simplicity, we consider only the branch depicted in figure 1. The reader can refer to [2] for a description of the different classes.

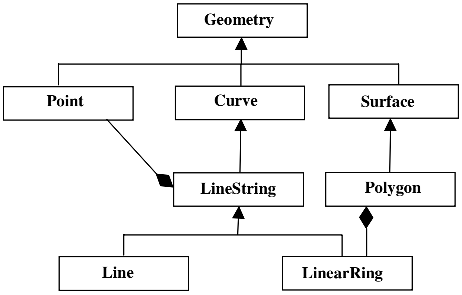


Fig. 1. OpenGIS Geometry Class Hierarchy

In section 2, we defined the set of objects *O* and the set of subjects *S*, with $S \subseteq O$ (any subject is also an object). We assume all objects in *O* to be geo-referenced. Therefore any object has some descriptive attributes and some spatial attributes and methods. These spatial attributes and methods can be used for specifying contexts, like any other attributes. For example, if *p* is an object whose geometry is a point then *x(p)* and *y(p)* represents its coordinates. If *l* is a Linestring then *pointN(l,3)* represents the third point of *l*…etc. If talking about the geometry of an entity is irrelevant then the geometry of this entity is the empty geometry ∅. Since subjects are mostly users, the geometry of subjects is generally a point. However, it could also be a polygon if the exact subject position cannot be determined precisely or should not be disclosed for privacy reasons. Location and/or shape of any object may change over time. This is obviously true for users whose coordinates are updated in real-time (thanks to GPS devices for example), but it can also be true for any other object.

3.2 Spatial Analysis Functions

Spatial analysis functions take one or more geometric objects as input and return either a number or another geometric object. [2] defines seven spatial analysis functions, namely *distance*, *buffer*, *convexHull*, *intersection* ($\cap$), *union* ($\cup$), *difference* (\) and *symDifference* (Δ). We add these spatial analysis functions to Or-BAC.

Let *a* and *b* be two geometric objects and *x* a scalar, these functions may be defined as follows:

- *distance (a, b)* – Returns the shortest distance (a scalar) between any two points in the two geometric objects *a* and *b*
- *buffer(a, x)* – Returns a geometric object that represents all points whose distance from geometric object *a* is less then or equal to *x*

- *convexHull(a)* – Returns a geometric object that represents the convex hull (mathematical definition) of geometric object *a*
- *a* $\cap$ *b, a* $\cup$ *b, a* \ *b, a* Δ *b*, – Respectively returns a geometric object that represents the point set intersection (resp. union, resp. difference, resp. symmetric difference) of object *a* with object *b*

We also add the following geometric functions to Or-BAC

- Let *a* be a geometric object, *I(a)*, *B(a)*, *E(a)* and *dim(a)* respectively returns the interior, boundary, exterior and dimension (-1 for the empty geometry Ø, 0 for `Point`, 1 for `Linestring` and 2 for `Polygon`) of *a*.

3.3 Spatial Predicates

Spatial predicates are used to test for the existence of a specified topological relationship between two geometric objects. [2] defines eight spatial predicates namely, *Equals*, *Disjoint*, *Intersects*, *Touches*, *Crosses*, *Within*, *Contains* and *Overlaps*.

We consider these predicates to be built-in Or-BAC predicates. Each predicate is defined over domains $O \times O$.

- $\forall g_1 \in O, \forall g_2 \in O, Equals(g_1, g_2) \leftrightarrow (g_1 \cap g_2) = (g_1 \cup g_2)$
- $\forall g_1 \in O, \forall g_2 \in O, Disjoints(g_1, g_2) \leftrightarrow g_1 \cap g_2 = \varnothing$
- $\forall g_1 \in O, \forall g_2 \in O, Touches(g_1, g_2) \leftrightarrow$ $(I(g_1) \cap I(g_2) = \varnothing) \wedge (g_1 \cap g_2 \neq \varnothing)$
- $\forall g_1 \in O, \forall g_2 \in O, Crosses(g_1, g_2) \leftrightarrow$ $(dim(I(g_1) \cap I(g_2)) < max(dim(g_1), dim(g_2)))$ $\wedge (g_1 \cap g_2 \neq g_1) \wedge (g_1 \cap g_2 \neq g_2)$
- $\forall g_1 \in O, \forall g_2 \in O, Within(g_1, g_2) \leftrightarrow$ $(g_1 \cap g_2 = g_1) \wedge (I(g_1) \cap E(g_2) \neq \varnothing)$
- $\forall g_1 \in O, \forall g_2 \in O, Contains(g_1, g_2) \leftrightarrow Within(g_2, g_1)$
- $\forall g_1 \in O, \forall g_2 \in O, Overlaps(g_1, g_2) \leftrightarrow$ $(dim(I(g_1)) = dim(I(g_2)) = dim(I(g_1) \cap I(g_2)))$ $\wedge (g_1 \cap g_2 \neq g_1) \wedge (g_1 \cap g_2 \neq g_2)$
- $\forall g_1 \in O, \forall g_2 \in O, Intersects(g_1, g_2) \leftrightarrow \neg Disjoint(g_2, g_1)$

3.4 Moving Objects

In [17], the authors emphasize the fact that there is a growing need for a movement-aware access control model. We show, in this section, how we can easily model security rules based on moving objects. Since software coupled with GPS devices are able to estimate the speed and the heading of an object, we assume that geo-referenced objects which can move have the two following velocity attributes:

- *speed* which indicates the speed of the object. The speed is a scalar value greater than or equal to 0.
- *direction* which indicates the direction taken by the object. The direction is an angle value between 0 and 360 degrees. It is equal to N/A (Not Applicable) if the speed is equal to 0.

These two velocity attributes can be used for specifying contexts, like any other attribute. For example, if *p* is a moving object then *speed(p)* and *direction(p)* respectively represents its speed and direction. Note that only moving objects or objects which can potentially move have these two velocity attributes. Objects which cannot move (like a tree or a house) do not have these two attributes. A moving object which has stopped its movement has a speed attribute value equal to 0 and a direction attribute value equal to N/A.

4 Modelling Spatial Contexts with Or-BAC

Spatial attributes and methods, geometric functions, spatial predicates and velocity attributes are the elementary bricks with which we can build spatial contexts. In this section, we try to figure out various types of spatial contexts (without pretending to be exhaustive) and we show how to model those contexts.

4.1 Spatial Contexts Related to Subject Position

Being able to express such contexts is critical in almost all kinds of Location Based Systems. Thanks to the primitives we defined in section 3 extending the first-order language of Or-BAC, we can easily express various conditions on the user position. Let us consider the following example:

Let *Firingzone* denote a geometric object whose geometry is a polygon corresponding to a zone from where recruits of the 1st Battalion have the permission to fire on some targets.

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(1^{st}\, Battalion, s, a, o, InFiringzone) \leftrightarrow \\ Within(s, Firingzone)$$

that is, at organization 1st Battalion, context *InFiringzone* holds between subject *s*, action *a* and object *o* if and only if subject *s* is within (polygon) *Firingzone*. The following example of permission uses context *InFiringzone*:

$$Permission(1^{st}\ Battalion,\ Recruit,\ Fire,\ Target,\ InFiringzone)$$

that is, recruits from the first battalion have the permission to fire on targets if and only if they are inside the firing zone. This security rule is based on the position of subjects belonging to role *Recruit*.

4.2 Spatial Contexts Related to Object Position

In the same way, thanks to the primitive defined in the previous section, we can easily define contexts related to object position. Consider the following example:

Let *Securityzone* denote a geometric object whose geometry is a polygon corresponding to a military security zone.

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(1^{st}\, Battalion, s, a, o, Intrusion) \leftrightarrow \\ Within(o, Securityzone)$$

that is, at organization 1st Battalion, context *Intrusion* holds between subject *s*, action *a* and object *o* if and only if object *o* is within (polygon) *Securityzone*. The following example of permission uses context *Intrusion*:

Permission(1st Battalion, Sentry, Arrest, Civilian, Intrusion)

That is, sentries from the first battalion have permission to arrest any civilian located within the security zone. This security rule is based on the position of objects belonging to view *Civilian.*

4.3 Spatial Contexts Related to Subject and Object Position

Mixing conditions on user position and object position can easily be done. Consider the following example:

$$\forall s \in S, \forall a \in A, \forall o \in O, \forall d \geq 0, Hold(1^{st}\, Battalion, s, a, o, Firingrange) \leftrightarrow \\ distance(s, o) \leq 0.5$$

That is, at organization 1st Battalion, context *Firingrange* holds between subject *s*, action *a* and object *o* if and only the distance between subject s and object o is less than or equal to 500 meters. The following example of permission uses context *Firingrange*:

Permission(1st Battalion, Artillery, Fire, Tanks, Firingrange)

That is, Artillery from the first battalion has the permission to fire on tanks which are within the range of 500 meters.

4.4 Geo-temporal Contexts

In [10], the authors define functions *before_time, after_time, before_date* and *after_date* which return a temporal context (see [10] for definition of these functions). For example, *after_time*(22:00) defines temporal context “after 22:00”. In [10], the authors also define the following operators for contexts composition:

- $\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall c_1 \in C, \forall c_2 \in C,$
  $Hold(org, s, a, o, c_1 \,\&\, c_2) \leftrightarrow Hold(org, s, a, o, c_1) \wedge Hold(org, s, a, o, c_2)$
- $\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall c_1 \in C, \forall c_2 \in C,$
  $Hold(org, s, a, o, c_1 \oplus c_2) \leftrightarrow Hold(org, s, a, o, c_1) \vee Hold(org, s, a, o, c_2)$

- $\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall c \in C,$
  $Hold(org, s, a, o, \bar{c}) \leftrightarrow \neg Hold(org, s, a, o, c)$

By composing temporal contexts and spatial contexts, we can define geo-temporal contexts. For example, let us first define context *Night* as follows:

$$Night = after\_time(22{:}00)\ \&\ before\_time(6{:}00)$$

By composing context Intrusion defined in section 4.2 and context Night, we can now define the following geo-temporal context:

$$Night\_intrusion = Intrusion\ \&\ Night$$

4.5 Spatial Contexts Related to Visualisation of Spatial Data

In some spatially aware access control models like GSAM, *zoom-in* is considered as a separate privilege like read or write. We prefer to model the zoom-in operation as a context of another operation, like the *display* operation for instance. We believe that this approach facilitates the interpretation of the security rules i.e. we cannot have conflicts between security rules regarding a display-like operation and some other security rules regarding the zoom-in operation.

Let *Defaultscale* be the default scale at which a geometric object is displayed. We define function *mzf* (maximum zoom-in factor) that takes as input a zoom-in factor and returns a spatial context[1] defined as follows:

$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall z \geq 1, Hold(org, s, a, o, mzf(z)) \\ \leftrightarrow scale(a) \leq (z \times Defaultscale)$$

that is, at organization *org*, context *mzf(z)* holds between subject *s*, action *a* and object *o* if and only if the *scale* parameter[2] of action *a* is less than or equal to the default scale *Defaultscale* multiplied by the zoom-in factor *z*.

The following example of permission uses context *mzf(2)*:

Permission(1st Battalion, Soldiers, Display, Barrack_Map, mzf(2))

That is, soldiers from the first battalion have the permission to display maps of barracks with a maximum zoom-in factor of 2.

4.6 Spatial Contexts Related to Movement

Since the velocity of a moving object is described in its speed and direction attributes, we can express spatial contexts related to movement like any other contexts. Consider the following example:

$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, Hold(org, s, a, o, Samevelocity) \\ \leftrightarrow speed(s) \leq (1.1 * speed(o)) \wedge (0.9 * speed(o)) \leq speed(s) \\ \wedge direction(s) \leq (5 + direction(o)) \wedge (direction(o) - 5) \leq direction(s)$$

[1] Defining a function returning a context is possible with Or-BAC. See [10] for the definition of functions *before_time* and *after_time* for instance.

[2] If action *a* does not have a scale parameter (descriptive attribute) then function *mzf* will never return any context.

that is, at organization *org*, context *Samevelocity* holds between subject *s*, action *a* and object *o* if and only if the *speed* of *s* is equal to the speed of *o* (± 10%) and the heading of *s* is the same as the heading of *o* (± 5 degrees). The following permission uses context *Samevelocity*

Permission(1st Battalion, Tank, Communicate, Tank, Samevelocity)

That is, tanks are allowed to communicate with each other provided they are moving in the same direction and at the same speed. This permission is an example of a security rule involving moving subjects and objects (where subjects are also objects).

5 Example of a Security Policy for a Spatial Application

Our language for defining spatial contexts allows us to define security policies which are based on the geometry of subjects and objects. The Or-BAC model extended to our language allows us to define security policies for any kind of geospatial application.

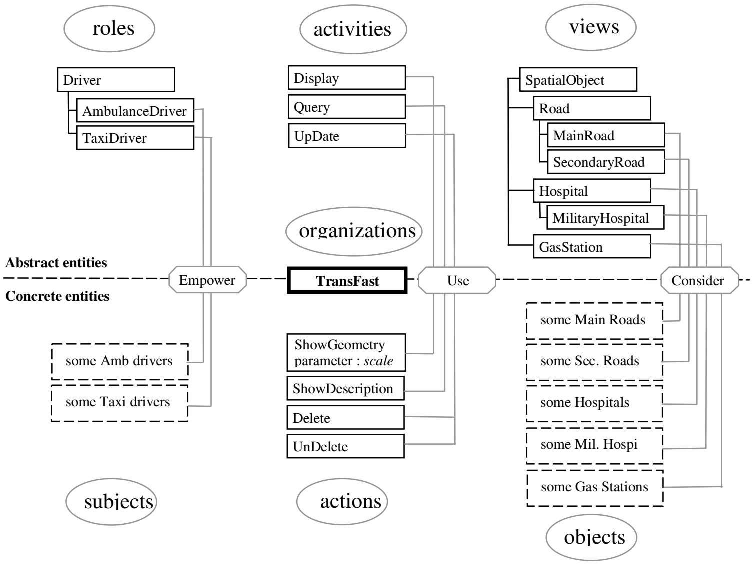


Fig. 2. Synopsis of our example

In this section, we consider an organization called *TransFast (TF)* simultaneously managing a fleet of taxis and a fleet of ambulances. While driving, drivers from this company use a spatial application displaying the road network (main and secondary roads), gas stations and hospitals. Drivers can also obtain descriptive data about gas station (opening hours …) and hospitals (number of available rooms …). Figure 2 shows the hierarchy and the relationships between abstract entities (roles, activities and views) and concrete entities (subjects, actions and objects). Note that action *ShowGeometry* (implements activity *Display*) for graphically displaying geometric objects has **one parameter** *scale* indicating the scale at which objects are displayed. Action *ShowDescription* (implements activity *Query*) can be used for displaying descriptive data of an object. Actions *Delete* and *UnDelete* (implement activity *Update*) allows to logically delete and undelete objects. We also define the constant *DefaultScale* giving the default scale at which a geometric object is displayed.

Furthermore, in order to prevent conflicts, we assume the following separation constraints: *separated_role(TF, TaxiDriver, AmbulanceDriver), separated_activity(TF, Query, Display), separated_activity(TF, Query, Update), separated_activity(TF, Update, Display), separated_view(TF, Road, Hospital), separated_view(TF, Road, GasStation), separated_view(TF, GasStation, Hospital)* and *separated_view(TF, MainRoad, SecondaryRoad)*

Finally, we define the following contexts:

- We define function *mzf* [3](maximum zoom-in factor) that takes as input a zoom-in factor and returns a spatial context defined as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, \forall z \geq 1, Hold(TF, s, a, o, mzf(z)) \\ \leftrightarrow scale(a) \leq (z \times Defaultscale)$$

  - that is, at organization *TF*, context *mzf(z)* holds between subject *s*, action *a* and object *o* if and only if the *scale* parameter of action *a* is less than or equal to the default scale *Defaultscale* multiplied by the zoom-in factor *z*.

- We define function *radius* that takes as input a distance and returns a spatial context defined as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, \forall d \geq 0, Hold(TF, s, a, o, radius(d)) \leftrightarrow \\ distance(s, o) \leq d$$

  - that is, context *radius(d)* holds if and only if the distance between subject *s* and object *o* is less than or equal to *d*.

- We define spatial context *On_theway* as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(TF, s, a, o, On\_theway) \\ \leftrightarrow \exists r, (Use(r, Road) \wedge Within(s, r) \wedge Touches(o, r))$$

  - that is, context *On_theway* holds if and only if object *o* touches the road *r* subject *s* is within

- We define temporal context *Rush_hours* as follows:

$$Rush\_hours = (after\_time(7:00)\ \&\ (before\_time(9:00)) \\ \oplus (after\_time(17:00)\ \&\ (before\_time(19:00))$$

[3] Although *mzf* has already been defined in section 4, we redefine it here but within the scope of organization *TF*.

- that is, context *Rush_hours* holds if and only if time is between 7:00 and 9:00 or 17:00 and 19:00

- We define temporal context *Not_moving* as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(TF, s, a, o, Not\_moving)$$
$$\leftrightarrow speed(s) = 0$$

  - that is, context *Not_moving* holds if and only if speed of subject s is equal to 0.

We can now express our security policy. Basically, the security policy expresses the fact that drivers can access to spatial data which are within a radius of 40km around their position. However, there are some restrictions to this general rule.

Rule 1: Drivers have the permission to display main roads at the default scale (i.e. with a maximum zoom-in factor of 1), whether these main roads are inside or outside the radius of 40km around the driver.

$$Permission(TF, Driver, Display, MainRoads, mzf(1))$$

Rule 2: Drivers have the permission to display with a maximum zoom-in factor of 5 any object that is within a radius of 40km around their position,

$$Permission(TF, Driver, Display, SpatialObjects, mzf(5)\ \&\ radius(40))$$

Note that Rule 1 and Rule 2 will be active at the same time if the subject plays role *Driver* (or a sub-role of role *Driver*), the action implements the *Display* activity, the object belongs to the view *MainRoad* and the maximum zoom-in factor is 1. However since Rule 1 and Rule 2 are both permissions, there is, of course, no conflict.

Rule 3: However, taxi drivers are prohibited do display secondary roads outside rush hours (this is because, outside rush hours, taxi drivers are supposed to use main roads only)

$$Prohibition(TF, TaxiDriver, Display, SecondaryRoads, \overline{Rush\_hours})$$

Rule 4: Moreover, for all drivers, gas stations which are not on the way cannot be displayed

$$Prohibition(TF, Driver, Display, GasStation, \overline{On\_theway})$$

Rule 5: Finally taxi drivers are prohibited to display military hospitals at a scale greater than the default scale, i.e. they are forbidden to zoom-in on a military hospital.

$$Prohibition(TF, TaxiDriver, Display, MilitaryHospital, \overline{mzf(1)})$$

Regarding descriptive data, the security policy states that drivers can query objects which are within a radius of 40km around their position:

Rule 6: Drivers can query any spatial object which is within a radius of 40km around their position.

$$Permission(TF, Driver, Query, SpatialObjects, radius(40))$$

Rule 7: However taxi drivers cannot query hospitals. Note that the context for this rule is the default context *Default_ctx* which is always true.

$$Prohibition(TF,TaxiDriver,Query,Hospital,Default\_ctx)$$

Regarding the update activity, the security policy states that drivers can delete a road if they detect that a road is blocked:

Rule 8: Drivers who have stopped their car can logically delete a road which is blocked (the road is then still displayed like any other road but appears with a different colour) or undelete a road if they detect that a road has been unblocked. The road has to be within a radius of 40km.

$$Permission(TF, Driver, Update, Road, radius(40)\ \&\ Not\_moving)$$

Regarding the above security policy, we can make the following comments addressing conflicts between rules:

Thanks to the separation constraints defined above (e.g. a subject cannot be empowered in both roles *TaxiDriver* and *AmbulanceDriver*), we avoid many potential conflicts. However, the following potential conflicts still remain:

- **Rule 3 and Rule 2:** There is a potential conflict between these two rules. Indeed these two rules lead to an actual conflict if the subject plays role *TaxiDriver* (sub-role of *Driver*), the action implements the activity *Display*, the object belongs to the view *SecondaryRoad*, the zoom-in factor is less than or equal to 5, the distance between the subject and the object is less than 40km and time is outside rush hours. We solve this conflict by assigning to Rule 3 a priority which is higher than the priority of Rule 2.
- **Rule 4 and Rule 2:** There is a potential conflict between these two rules. Indeed these two rules lead to an actual conflict if the subject is a *Driver*, the action implements the activity *Display*, the object belongs to the view *GasStation*, the zoom-in factor is less than or equal to 5, the distance between the subject and the object is less than 40km and the object is not on the way. We solve this conflict by assigning to Rule 4 a priority which is higher than the priority of Rule 2.
- **Rule 5 and Rule 2:** There is a potential conflict between these two rules. Indeed these two rules lead to an actual conflict if the subject plays role *TaxiDriver*, the action implements the activity *Display*, the object belongs to the view *MilitaryHospital*, the distance between the subject and the object is less than 40km and the zoom-in factor is strictly greater than 1. We solve this conflict by assigning to Rule 5 a priority which is higher than the priority of Rule 2.
- **Rule 7 and Rule 6:** There is a potential conflict between these two rules. Indeed these two rules lead to an actual conflict if the subject plays role *TaxiDriver*, the action implements the activity *Query*, the object belongs to the view *Hospital* and the distance between the subject and the object is less than 40km. We solve this conflict by assigning to Rule 7 a priority which is higher than the priority of Rule 6.

6 Comparison with Related Works

During the last decade there has been a significant increase in the amount of papers dealing with access control to geographic data [18] [19] [20] [21] [22] [23] [24] [9]. Due to space limitations, we focus on the two following authorization models which we consider to be the two most advanced models for controlling access to spatial data: Geo-RBAC [24] and GSAM (GeoSpatial Authorization Model) [9].

The Geo-RBAC model was introduced for the first time in [8] then further formalized and extended in [24]. Geo-RBAC defines the concept of *spatial role*. A spatial role is a pair <*r,e*> where *r* is a traditional role and *e* the spatial extent of the role. The role extent defines the boundaries of the space in which the role can be assumed by the user. The GeoSpatial Authorization Model (GSAM) was one of the first models dealing with access control systems for geographical data and was first proposed in [25], then successively extended in [26], [27] and [9]. In [9], the authors introduce the concept of *geo-temporal role*. A geo-temporal role is a pair <*r,sc*> where *r* is a traditional role and *sc* a *scene*. The concept of scene resembles the concept of spatial extent in Geo-RBAC except that a scene includes a temporal extent. We believe that the Or-BAC model has many advantages over these two models. The main reasons are the followings:

- Although Geo-RBAC and GSAM borrow the concept of role from the traditional RBAC model, they are *ad hoc* models more or less designed for a particular type of geospatial application. Geo-RBAC is mainly for applications requiring location based access controls whereas GSAM is mainly for regulating access to satellite images. Consequently, none of these models has the flexibility of the Or-BAC model, which is a generic authorisation model.
- Neither Geo-RBAC nor GSAM formalizes the notion of context. Instead of defining spatial contexts, these two models define the rather non-intuitive concept of spatial role to be able to include spatial conditions in the security policy. Spatial roles are mainly for expressing spatial conditions on subjects. Consequently, Geo-RBAC and GSAM fail at expressing spatial conditions on objects (unless the objects are themselves subjects). In fact, many of the contexts we described in this paper cannot be expressed neither with Geo-RBAC nor with GSAM.
- Another important drawback of spatial (or geo-temporal) roles is that they greatly increase the number of roles in the role hierarchy making the management of the security policy more complex.
- The last reason (but not the least) is that the Or-BAC model including our spatial primitives for specifying spatial contexts inherits the qualities of the core Or-BAC model. For instance, we benefit from the useful and innovative concepts of activity and view for structuring the security policy and we benefit from the conflict resolution strategy of the Or-BAC model[4]. Conflict resolution is barely addressed in both Geo-RBAC and GSAM whereas in [14], the authors show that conflict detection in Or-BAC is tractable in polynomial time.

[4] In [14], conflict detection deals with conflicts between permissions and prohibitions. Analyzing conflicts between prohibitions and obligations remains work to be done.

7 Conclusion

In this paper we showed how to model dynamic security rules based on spatial contexts. We extended the core Or-BAC first order logic language for representing contexts with some OGC compliant spatial functions and predicates. We introduced a typology of spatial contexts: spatial contexts based on the position of subjects and/or objects, geo-temporal contexts, contexts based on the movement of objects and/or subjects and contexts focusing on the visualization of spatial data. Through a real life application, we showed how we can easily express various kind of dynamic security rules based on some complex geo-temporal contexts. Finally we showed that many spatial contexts cannot be easily handled by existing spatially-aware models like Geo-RBAC or GSAM. Future works shall include the followings:

- Designers of the Or-BAC model have developed MotOrBAC [28]. MotOrBAC is a security policy tool which can be used to specify, simulate and administrate security policies. MotOrBAC has been developed on top of the Or-BAC application programming interface (API), a java API. MotOrBAC uses the Jena inference engine [29] for deriving conflicts and concrete rules from abstract rules. We plan to extend MotOrBAC with the proposed spatial functions and predicates in order to simulate spatially driven security policies.
- Complex spatial contexts may lead to situations where users may face difficulties to know where and when they can be granted the permission to access to a given resource. We plan to define methods for automatically answering questions like "Where should I be located to gain access to this object ?" or "Show me on the map which objects can be accessed according to my current position"
- In Destination Moon [30], Tintin has the permission to destroy the X-FLR6 rocket prototype if it veers off its *trajectory*. In [31], the authors define the concept of trajectory as a sequence of moves and stops enriched with some semantic annotations allowing users to attach semantic data to specific part of the trajectory. They also propose solutions for modelling trajectories. Based on their work, we plan to extend the Or-BAC language with primitives allowing us to write trajectory-aware security rules.
- In this paper, we limited ourselves to a two dimensional geometric model. We shall consider in future works a three dimensional geometric model.
- Finally, we plan to investigate the emerging concept of spatial/location privacy protection policies.

References

1. El-Kalam, A., El-Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miège, A., Saurel, C., Trouessin, G.: Organization Based Access Control. In: 4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy 2003). IEEE, Como (2003)
2. Herring, J.R.: OpenGIS(R) Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture. Open Geospatial Consortium Inc. OGC(R) 06-103r3 (2006)
3. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. IEEE Computer 29, 38–47 (1996)

4. Moyer, M., Ahamad, M.: Generalized Role-Based Access Control. In: Proceedings of the 21st International Conference on Distributed Computing Systems. IEEE Computer Society, Los Alamitos (2001)
5. Park, S.-H., Han, Y.-J., Chung, T.-M.: Context-Role Based Access Control for Context-Aware Application. In: Gerndt, M., Kranzlmüller, D. (eds.) HPCC 2006. LNCS, vol. 4208, pp. 572–580. Springer, Heidelberg (2006)
6. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A temporal role-based access control model. ACM Transactions on Information and System Security (TISSEC 2001) 4, 191–233 (2001)
7. 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, 4–23 (2005)
8. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: A spatially Aware RBAC. In: ACM Symposium on Access Control Models and Technologies (SACMAT 2005), Stockholm, Sweeden, pp. 29–37 (2005)
9. Atluri, V., Chun, S.A.: A geotemporal role-based authorization system. International Journal of Information and Computer Security 1, 143–168 (2007)
10. Cuppens, F., Miège, A.: Modeling Contexts in the Or-BAC Model. In: 19th Annual Computer Security Applications Conference (ACSAC 2003), Las Vegas, NV, USA (2003)
11. Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual security policies. International Journal of Information Security (IJIS 2008) 7, 285–305 (2008)
12. OGC: Open Geospatial Consortium Inc. - About Us (2008)
13. Cuppens, F., Cuppens-Boulahia, N., Miège, A.: Inheritance hierarchies in the Or-BAC model and application in a network environment. In: Second Foundation of Computer Security WorkShop (FCS 2004), Turku, Finland (2004)
14. Cuppens, F., Cuppens-Boulahia, N., Ghorbel, M.B.: High Level Conflict Management Strategies in Advanced Access Control Models. Electronic Notes in Theoretical Computer Science (ENTCS) 186, 3–26 (2007)
15. Janée, G., Frew, J., Hill, L.L.: Issues in Geo-referenced Digital Libraries. D-Lib Magazine 10 (2004)
16. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with application to GIS. Elsevier, Amsterdam (2002)
17. Damiani, M.L., Silvestri, C.: Towards movement-aware access control. In: ACM SIGSPATIAL GIS 2008 International Workshop on Security and Privacy in GIS and LBS (SPRINGL2008), pp. 39–45. Association for Computing Machinery, Irvine (2008)
18. Bertino, E., Damiani, M.L., Momini, D.: An access control system for a Web map management service. In: Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications (RIDE 2004), pp. 33–39 (2004)
19. Belussi, A., Bertino, E., Catania, B., Damiani, M.L., Nucita, A.: An Authorization model for geographical maps. In: Proceedings of the 12th annual ACM International Workshop on Geographic Information Systems (RIDE 2004), Washington DC, USA, pp. 82–91 (2004)
20. Hansen, F., Oleshchuk, V.: SRBAC: A spatial Role-Based Access Control Model for Mobile Systems. In: 7th Nordic workshop on secure IT systems (NORDSEC 2003), Gjvik, Norway, pp. 129–141 (2003)
21. Matheus, A., Herrmann, J.: Geospatial eXtensible Access Control Markup Language (GeoXACML). Open Geospatial Consortium Inc. OGC(R) 07-026r2 (2008)

22. Gabillon, A., Capolsini, P.: DRM policies for Web Map Service. In: ACM SIGSPATIAL GIS 2008 International Workshop on Security and Privacy in GIS and LBS (SPRINGL 2008), pp. 20–29. Association for Computing Machinery, Irvine (2008)
23. Chandran, S.M., Joshi, J.B.D.: LoT-RBAC: A location and time-based RBAC model. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 361–375. Springer, Heidelberg (2005)
24. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: GEO-RBAC: A spatially Aware RBAC. ACM Transactions on Information Systems and Security, 1–34 (2006)
25. Chun, S.A., Atluri, V.: Protecting privacy from continuous high-resolution satellite surveillance. In: Proceedings of the 14th IFIP 11.3 Annual Working Conference on Database Security, Schoorl, The Netherlands, pp. 233–244 (2000)
26. Atluri, V., Mazzoleni, P.: A uniform indexing scheme for geo-spatial data and authorizations. In: Proceedings of the 16th IFIP WG 11.3 Conference on Data and Application Security (2002)
27. Atluri, V., Chun, S.A.: An authorization Model for Geospatial Data. IEEE Transactions on Dependable and Secure Computing 1, 238–254 (2004)
28. Autrel, F., Cuppens, F., Cuppens-Boulahia, N., Coma, C.: MotOrBAC 2: a security policy tool. In: 3rd Conference on Security in Network Architectures and Information Systems (SAR-SSI 2008), Loctudy, France, pp. 273–288 (2008)
29. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: Implementing the semantic web recommendations. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, USA, pp. 74–83 (2004)
30. Hergé: Destination Moon (The adventures of Tintin). Casterman (1953)
31. Spaccapietra, S., Parent, C., Damiani, M.L., Macedo, J.A.d., Porto, F., Vangenot, C.: A Conceptual View on Trajectories. Data and Knowledge Engineering 65, 124–146 (2008)

Medical Image Integrity Control Combining Digital Signature and Lossless Watermarking

Wei Pan[1,3], Gouenou Coatrieux[1,3], Nora Cuppens-Boulahia[2,3],
Frederic Cuppens[2,3], and Christian Roux[1,3]

[1] Institut Telecom; Telecom Bretagne; Unite INSERM 650 LaTIM,
Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3 France
{wei.pan,gouenou.coatrieux,christian.roux}@telecom-bretagne.eu

[2] Institut Telecom; Telecom Bretagne; LUSSI Department, 2 rue de la Châtaigneraie,
CS 17607, 35576 Cesson Sévigné Cedex France
{nora.cuppens,frederic.cuppens}@telecom-bretagne.eu

[3] Université Européenne de Bretagne, France

Abstract. Enforcing protection of medical content becomes a major issue of computer security. Since medical contents are more and more widely distributed, it is necessary to develop security mechanism to guarantee their confidentiality, integrity and traceability in an autonomous way. In this context, watermarking has been recently proposed as a complementary mechanism for medical data protection. In this paper, we focus on the verification of medical image integrity through the combination of digital signatures with such a technology, and especially with Reversible Watermarking (RW). RW schemes have been proposed for images of sensitive content for which any modification may affect their interpretation. Whence, we compare several recent RW schemes and discuss their potential use in the framework of an integrity control process in application to different sets of medical images issued from three distinct modalities: Magnetic Resonance Images, Positron Emission Tomography and Ultrasound Imaging. Experimental results with respect to two aspects including data hiding capacity and image quality preservation, show different limitations which depend on the watermark approach but also on image modality specificities.

1 Introduction

With the advances of Internet technology, especially in healthcare, images can be cross-exchange in right time allowing new medical practice through for example telediagnosis, teleconsultation services. At the same time, ensuring the security of exchanged medical data becomes a major issue. Three mandatory characteristics need then to be addressed: confidentiality, availability and reliability based on the outcomes of information integrity and authenticity.

Current healthcare information systems are no longer based on a centralized architecture introducing the need for means to control distribution of medical contents in distributed infrastructures. Enforcing content protection using classical access control mechanisms is no longer sufficient. It is thus necessary to

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 153–162, 2010.

develop security mechanisms that guarantee protection of medical contents in an autonomous way, especially their integrity and traceability.

In such a framework, watermarking has been shown as a complementary mechanism to enhance medical image security [1] [2]. In general speaking, Watermarking allows inserting a message, also called a watermark, in a host document by modifying the host content in an imperceptible way. For one image, the message is attached at the signal level slightly modifying its gray values. Whence, the hosted message and the host image are intimately associated independently of the image file format. By its ability to introduce a protection level the nearest as possible of the data, watermarking can rise up medical image reliability by asserting its integrity and its authenticity (i.e. an evidence that the information belongs to the correct patient and is issued from the right source). To do so, the embedded message may for instance correspond to a digital signature of the image pixels [3] [4].

For medical images, it is widely expected that the watermark should not hinder the qualitative perception of the image. This constraint implies that the interpretation of the image by a specialist shall remain unchanged after message insertion. However, the majority of watermarking methods irreversibly alters the image. Distortions may be low-level when the watermark insertion is weighted by use of a visual perception model [5], but to our knowledge none of these models has been validated in the case of medical imaging. Consequently, these distortions may mask some subtle image details.

Reversible or lossless watermarking has been proposed to overcome this issue. It allows the user to reconstruct the original image after having extracted the watermark (i.e. by removing image distortion). However, once the watermark has been removed, the image is no more protected, just like for data encryption. So even if removing the watermark is possible, most applications have a high interest to keep it as long as possible in the image in order first to continuously protect the information and second to not limit image interpretation to compliant systems (i.e. with watermarking abilities). Whence, in our view, even for reversible watermarking, the imperceptibility property has to be guaranteed in the medical domain. The reversible property has an interest for watermark content update.

Several reversible watermarking methods have been proposed since 1999. We have selected 13 the most representative methods [8-20] to give a classification in section 3. They introduce more or less visible distortions with varying insertion capacities. Capacity is the amount of information that can be embedded into one image and which is expressed in bit of message per pixel of image (*bpp*). In this paper, we have tested some of these methods among different medical image issued from different modalities (MRI (magnetic resonance imaging), PET (positron emission tomography) and US (ultrasound imaging) for the purpose of verifying the integrity of medical images by embedding a digital signature. Before comparing these methods with respect to the criterions given above in section 4, we present in section 2 an integrity control verification process based on lossless watermarking and cryptographic hash. Conclusions are made in section 5.

2 Verifying Integrity of Medical Image with Lossless Watermarking

Integrity control of images can be addressed at two levels, that is: strict integrity control whereby one has to guarantee that the whole image is preserved as entire bit planes, or; content-based control in which pixels are allowed to vary while the visual content meaning remains preserved. In this work our interest is given to strict integrity which can be achieved by making use of cryptographic hash function.

Cryptographic hash functions are commonly used for digital signatures as they extract a resume or digest from the message data to be protected. Between the two function classes, the first one, called Message Code Authentication (MCA), uses a secret key and permits signature identification. The second one, known as Manipulation Detection Code (MDC), is calculated without a secret key. Since MCA function usually makes use of a MDC function concatenated with a secret key or asymmetrically encrypted, interest is given here to MDC hash function. These functions are said one way hash functions (i.e. non reversible), and from a message of arbitrary length they provide a fixed length digest or resume. For example, one of the best known methods is the SHA-256 (Secure Hash Algorithm) that yields to a signature of 256 bits [6]. Its collision probability, that is the probability to find another message with the same hash, is upper bounded by $1/2^{256}$. SHA also has good dispersion property in that a slight difference in a message will lead to a very different signature.

Such a cryptographic hash can be encrypted in asymmetric way allowing non repudiation property. The RSA (Rivest Shamir Adleman) algorithm [7] is the most widely-used asymmetric system. The system uses two different keys for encryption and decryption. One of these two keys, the public key, is meant to be known to everyone, and the other, the private key, is known to only one individual. In order to write to a recipient, all that needs to happen is to encrypt the message with the public key of the recipient. Upon reception, only the recipient will be able to decrypt the message with his private key. Data confidentiality is ensured in that case. The RSA algorithm allows also encryption with ones own private key (signature). In this case, everyone can read the message thanks to the public key. Since the sender is potentially the only person who could have encrypted it with his private key: the sender has signed the message. In DICOM (Digital Imaging and COmmunications in Medicine), the standard of reference for medical image storage and sharing (medical.nema.org), there exists a digital signature profile based on the RSA. This profile is combined with the RIPEMD-160, MD5, or SHA-1 hashing functions to generate a MAC (Message Authentication Code), which is encrypted using a private RSA key. This digital signature is actually stored in the header of a DICOM image file.

Reversibly watermarking a cryptographic hash within a medical image leads to the integrity control process illustrated in Fig. 1. A hash of the image I to be protected is calculated making use of a cryptographic hash function H ($H(I)$) and is then embedded in I leading to the watermarked image I_w. At the verification stage, the watermark reader extracts the hash $H(I)$ and removes

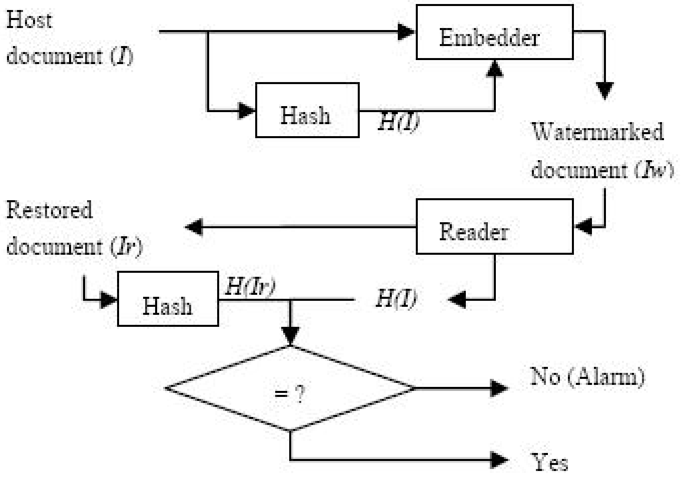


Fig. 1. Verifying image integrity thought reversible watermarking and cryptographic hash function

the watermark from I_w obtaining the restored image I_r. $H(I)$ is compared to $H(I_r)$. If $H(I)$ and $H(I_r)$ are equal then I_r is said to be identical to I; if not, the system states that the image has been modified. The hash can be calculated on the image pixel gray values or on the full representation of the document. In the latter case the integrity will also depend on the image file format.

With such a system, any modifications will give an alarm. However, the reversibility property allows the hash update like in the case of an authorized image modification, like a lossy image compression.

Several lossless watermarking schemes have been proposed in the literature. Each of them allows the reversible embedding of a message within an image while inducing at the same time more or less visible distortions. In the next section, we compare these different methods for different medical image modality.

3 Lossless Watermarking Methods

Two classes of reversible watermarking methods may be distinguished: additive methods and substitutive methods.

3.1 Additive Schemes

In the case of an additive insertion, the message m to be embedded is first transformed into a watermark signal w, next added to the host signal s leading to the watermarked signal sw: $s_w = s + w$.

Additive insertion has been primarily applied in the spatial domain in which the image pixel gray level values are limited to a fixed dynamic (2^p possible gray levels for an image of p bits depth). Consequently, watermark addition may lead to over/underflows, it means that modified pixel values may fall out of the allowed gray value range [0...2^p-1]. Obviously, such a problem occurs also when embedding is conducted in a transformed domain like in the wavelet or DCT domain.

Different strategies have been proposed to overcome over/underflow problem. One approach introduced in [8] consists in using modulo arithmetic. Insertion equation $I_w = (I + w) \bmod 2^p$ can however lead to a salt and pepper noise due to jumps between congruent values of the dynamic. An improved version of this method has been proposed in [13] where visual distortions are minimized by making use of arithmetic modulo on shorter cycles, obtained by splitting the signal dynamic in ranges of small size.

Another approach makes use of a signal classification before message embedding. In [9], the proposed scheme is based on image signal estimation, an image of reference invariant to the insertion process. More clearly the image and its watermarked version will have the same image of reference. In a first time, the reference image is used to decide whether or not a pixel block can be modified. The image of reference serves a classification procedure for identifying blocks that if modified lead to an over/underflow. Then insertion is conducted on the authorized parts of image by modulating the difference between the original image and its estimated version. As the image of reference is the same for the watermarked image, the decoder can easily retrieve watermarked parts of the image.

A third approach regroups methods that modulate the image histogram in a spatial or transformed domain. The method suggested by Ni *et al.* in [10] shifts a range of the image histogram. This range is identified by the couple (zp, pp), where zp and pp correspond respectively to the gray levels with the smallest ("zero-point") and the highest ("peak-point") number of pixels. This range is shifted by adding or subtracting one gray level from the peak point toward the zero point in order to leave one gray level (a "gap") near the peak point empty. Pixels that belong to the peak point class are moved to the gap or left unchanged for message embedding. Two gray values are used to code the message. Consequently, the alteration is not more important that one gray level for the modified pixels. However, the embedded data cannot be recovered unless the position of initial peak point is known by the decoder. This modulation has been applied in the wavelet domain by Xuan *et al.* [11] where the identification of the couple (zp, pp) is simplified as integer wavelet coefficients have a "laplacian" distribution centered around '0'.

Leest *et al.* [12] have proposed a similar approach. This latter is based on creating "gaps" at the minimum and maximum luminance values in local histograms of 2 × 2 pixels blocks. However with this approach, positions of pixels which have the value 0 and 2^p-1 have to be embedded in the image to solve the over/underflow problem. As a consequence, embedding capacity decreases when the numbers of such a pixel increase.

3.2 Substitutive Schemes

Substitutive insertion technique differs from the additive in the sense that rather than disrupting the signal by adding a watermark, it comes directly to replace the signal by another one stemmed from a predetermined dictionary signal. For example: the basic LSB scheme removes the pixels' least significant bits by bits of the message to be embedded. To make this scheme reversible, original binary values should be preserved and communicated to the decoder. Fridrich *et al.* [13] have shown that there exists a bit-plane B in the original image I, so that B can be losslessly compressed and disrupted randomly, without visible distortion in I. If such a bit-plane exists, it can be replaced by its compressed version and a binary message m. The insertion capacity of such a method is $|B|$-$|compress(B)|$ bits, where $|.|$ denotes the cardinal. Since several solutions have been proposed, some do not required embedding of data overhead. We class them into two categories: Lossless Compression Embedding (LCE) techniques and Expansion Embedding (EE) techniques.

Xuan *et al.* have proposed an insertion technique on coefficients of the integer wavelet transform [14]. They losslessly compress one or more middle bit-planes of integer wavelet coefficients to save space for data embedding. Celik *et al.* [15] proposed a generalized LSB substitutive technique, which firstly converts the binary message ($w \in \{0, 1\}$) to M-ary watermark ($w \in \{0, 1, , M - 1\}$) by arithmetic coding. For example, a watermark w can be converted from $(1000101011)_2$ to $(4210)_5$, where $M = 5$. Then the lowest M-levels of the pixels of the original image are replaced by the M-ary watermarks: $p_w = M\lfloor p/M \rfloor + w$, where p and p_w represent the original pixel and its watermarked version respectively and, $\lfloor . \rfloor$ the "floor" operator meaning "the greatest integer less than or equal to". The original values are losslessly compressed using the CALIC algorithm [21].

Differently to the above-mentioned LCE techniques, Tian's algorithm [16] may be the first one to use the Expansion Embedding technique for reversible watermarking. EE shifts to the left the binary representation of an integer value h to watermark (h can be a gray value or a transformed coefficient), thus creating a new virtual LSB that can be used for insertion: $h_w = 2h + b$, where h_w is a watermarked value and b is one bit of the message. To control the insertion distortion, the EE is combined with LSB substitution: $h_w = 2\lfloor h/2 \rfloor + b$. LSB substitution is applied to h values which cannot be expanded because of the limited dynamic of the signal or because of the limited distortion to be applied. As LSB substitution is used, original LSBs have to be watermarked along with the message. To distinguish at the reader stage which h values have been expanded, a binary location map L is required. In Tian's scheme L is losslessly compressed and added to the embedded message with the original LSBs. Alattar extended this scheme by applying the EE to a generalized integer transform [17]: several bits are embedded into vectors of adjacent pixels.

In the same way, Lee *et al.* [18] divide a pixel image into 16×16 pixel blocks, and a watermark is embedded into the high-frequency wavelet coefficients of each block by LSB-substitution or EE technique. Their location map is of small dimension ($(M \times N)/(16 \times 16)$) and does not require to be compressed. Always in the same view, Xuan *et al.* in their scheme [19] introduce a threshold T. If the absolute value of an integer wavelet coefficient is lower than T, then EE is applied for data embedding. With this approach, it may be difficult for the reader to distinguish between watermarked and non-watermarked coefficients. To solve this problem, the coefficients which have the absolute values higher or equal to T should be shifted to the left or right according to their signs by $T-1$ or T. So all watermarked coefficients that carry the message are in the interval $]-2T+1, 2T[$. With this approach there is no need for a location map. This is almost the same for the method proposed by Thodi *et al.* [20], which combines Tian's method and this shifting pretreatment in order to gain better performances.

All of these methods are known to be fragile, i.e. the watermarks will not survive any image alteration. This is why these methods are at first proposed for data integrity control. For this study, we have implemented some of the most recent or original methods, and indicated by their authors as efficient on usual test images such as "Lena", "Baboon" .... Three of these schemes are additive: Ni *et al.* [10], Leest *et al.* [12], Coatrieux *et al.* [9] and two substitutive: Xuan *et al.* [19], Thodi *et al.* [20].

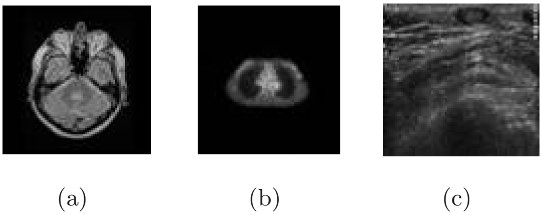
4 Losslessly Watermarking Medical Images

The five algorithms were implemented with MATLAB and the message bits were generated by the function of the MATLAB rand(). Experiments were conducted on three modalities: three 12 bits encoded MRI volumes of 79, 80 and 99 axial slices of 256 × 256 pixels respectively, three 16 bits encoded PET volumes of 234, 213 and 212 axial slices of 144 × 144 pixels respectively, and, three sequences of 8 bits encoded ultrasound images (14 of 480 × 592 pixels, 9 and 30 of 480 × 472 pixels respectively). Fig. 2 gives some samples of our data set.

To objectively quantify algorithms' performances, two indicators have been considered: the capacity rate C expressed in *bpp* and, in order to quantify the distortion between an image I and its watermarked version I_w, the peak signal to noise ratio (PSNR):

$$PSNR = 10log_{10}(\frac{NM(2^p-1)^2}{\sum_{i,j=1,1}^{N,M}(I(i,j)-I_w(i,j))^2}) \qquad (1)$$

where p corresponds to the image depth, N and M correspond to the image dimensions.



(a) (b) (c)

Fig. 2. Image samples extracted from our test set (a) MRI of the head-axial slice of 256 × 256 pixels, 12 bits encoded. (b) PET image of 144 × 144 pixels, 16 bits encoded, (c) ultrasound image of 480 × 592 pixels encoded on 8 bits.

Results are given in Tables 1 and 2. They provide the mean value and the standard deviation of the capacity and of distortion for each method and image modality. If we consider additive schemes in Table 1, [10] and [12] allow a watermark capacity close to 0.2 *bpp* with PSNR about 73-75 *dB* for MRI, 97-99 *dB* for PET. This means that nearly 13000 bits can be embedded in MRI slice and 4000 bits within PET slice. It is almost the same for ultrasound images ($>$ 10000 bits). [9] provides higher capacity for ultrasound images but may failed to watermark MRI slice as the capacity is rather small.

Results of substitutive methods [19] [20] are less effective than for additive methods [10] [12] when considering MRI and PET modalities. On the contrary, for ultrasound images, these methods are more efficient than additive methods. However, for the minimal distortion (see Table 2), the smallest attended capacity is greater than 1000 bits which is enough in our framework. For ultrasound images in Table 2, [19] and [20] propose a compromise of 0.14 *bpp*/48.77 *dB* and 0.22 *bpp*/48.44 *dB* respectively. Even if some methods keep limited as they require embedding a lot of information for reconstructing the original image along with the message, it is possible to embed one digital signature. However

Table 1. Capacity and distortion measurements for additive methods: Ni *et al.* [10], Leest *et al.* [12] and Coatrieux *et al.* [9]. Standard deviation is given in parenthesis.

	MRI		PET		US	
	C(*bpp*)	PSNR(*dB*)	C(*bpp*)	PSNR(*dB*)	C(*bpp*)	PSNR(*dB*)
[10]	0.26(0.011)	73.00(0.46)	0.20(0.013)	97.98(0.92)	0.05(0.053)	52.63(4.19)
[12]	0.20(0.007)	75.72(0.067)	0.22(0.033)	99.57(0.29)	0.04(0.013)	53.19(0.52)
[9]	0.0031(0.002)	78.43(0.84)	0.020(0.016)	100.79(1.16)	0.101(0.032)	48.51(0.20)

Table 2. Capacity and distortion measurements for MRI image axial slices, PET axial slices and ultrasound images. Standard deviation is given in parenthesis.

	Thodi *et al.* [20]		Xuan *et al.* [19]	
	C(*bpp*)	PSNR(*dB*)	C(*bpp*)	PSNR(*dB*)
MRI	0.021(0.004)	72.40(0.17)	0.098(0.012)	68.84(0.068)
	0.199(0.015)	44.62(3.44)	0.02(30%)	65.47(30%)
PET	0.13(0.026)	97.27(0.30)	0.15(7%)	93.73(7%)
	0.212(0.03)	67.87(2.77)	0.31(2%)	90.51(2%)
US	0.22(0.090)	48.44(0.77)	0.14(0.012)	48.77(0.65)
	0.49 (0.02)	40.58(2.88)	0.55(0.02)	43.22(0.60)

it must be noticed that for images, [19] was not able to insert a message as the amount of information for reconstruction was more important than the offered capacity. For PET images, in Table 2, only 7% of 659 images can be watermarked with a compromise C/PSNR = 0.15 *bpp*/93.73 *dB*. Considering the integrity control process shown in section 2 - Fig. 1, most methods allow the embedding of one hash produced by the SHA-256 hash function. With such a hash length of 256 bits, if we consider the constraint of preserving the image quality at best, [12] seems to be the most adapted. When the question is to protect the whole image volume, [9] will be more appropriate. Beyond integrity control, if the objective is the insertion of a big amount of information: [10] offers a compromise of 0.26 *bpp*/73 *dB* for MRI, [12] proposes 0.22 *bpp*/99.57 *dB* for PET and at least, for ultrasound images, [19] proposes a compromise of approximately 0.55 *bpp*/43.3 *dB*. Regardless the medical image modality, [20] proposes a satisfactory compromise of 0.021 *bpp*/72.40 *dB*, 0.13 *bpp*/97.27 *dB* and 0.22 *bpp*/48.44 *dB* for MRI, PET and ultrasound images respectively.

5 Conclusion

The main advantage of watermarking technology is to provide an autonomous and continuous protection of contents. In medical imaging, watermarking allows different applications. Also the performances of the proposed solutions vary according to the method proposed. Reversible watermarking is of main concern for medical images. However, in order to beneficiate of the watermarkings advantages, it is mandatory to propose reversible methods which minimize distortion and maximize capacity.

In this article, five reversible watermarking methods have been implemented and compared under different imaging modalities for the purpose of verifying the integrity of medical images though cryptographic hash embedding. Some limitations have been identified. They are mainly related to specific imaging modalities for which each method gives variable results in terms of capacity and distortion. From these experiments, it appears that the methods [12] are more suitable for PET, MRI and ultrasound images since they allow signature insertion with the smallest distortion.

Based on the presented work, the optimization is to modify the studied methods taking into account the specificities of the signal to be watermarked. Beyond verifying the integrity of medical images, there is a need for inserting a significant amount of data in order to cover a wide field of applications ranging from data protection (integrity, authenticity, traceability) to the addition of metadata.

References

1. Coatrieux, G., Maître, H., Sankur, B., Rolland, Y., Collorec, R.: Relevance of Watermarking in Medical Imaging. In: Proc. of IEEE EMBS Int. Conf. ITAB, Arlington, USA, pp. 250–255 (2000)
2. Zhou, X.Q., Huang, H.K., Lou, S.L.: Authenticity and integrity of digital mammography images. IEEE Trans. on Medical Imaging 20(8), 784–791 (2001)

3. Chao, H.M., Hsu, C.M., Miaou, S.G.: A Data-Hidding Technique With Authentication, Integration, and Confidentiality for Electronic Patient Records. IEEE Trans. on Information Technology in Biomedicine 6(1), 46–53 (2002)
4. Coatrieux, G., Lecornu, L., Sankur, B., Roux, C.: A Review of Image Watermarking Applications in Healthcare. In: Proc. of the IEEE EMBC Conf., New York, USA, pp. 4691–4694 (2006)
5. Piva, A., Barni, M., Bartolini, F., Capellini, V.: Exploiting the cross-correlation of RGB channels for robust watermarking of color images. In: Proc. of IEEE Int. Conf. on ICIP, vol. I, pp. 306–310 (1999)
6. Gilbert, H., Handschuh, H.: Security Analysis of SHA-256 and Sisters. In: Selected Areas in Cryptography, pp. 175–193 (2003)
7. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
8. Honsinger, C.W., Jones, P., Rabbani, M., Stoffel, J.C.: Lossless recovery of an original image containing embedded data. US Patent application, Docket No.:77102/E-D (1999)
9. Coatrieux, G., Lamard, M., Daccache, W., Puentes, J., Roux, C.: A low distortion and reversible watermark application to angiographic images of the retina. In: Proc. of the IEEE EMBC Conf., Shanghai, China, pp. 2224–2227 (2005)
10. Ni, Z., Shi, Y., Ansari, N., Wei, S.: Reversible data hiding. In: Proc. IEEE Int. Symp. Circuits and Systems, May 2003, vol. 2, pp. 912–915 (2003)
11. Xuan, G., Yao, Q., Yang, C., Gao, J., Chai, P., Shi, Y.Q., Ni, Z.: Lossless Data Hiding Using Histogram Shifting Method Based on Integer Wavelets. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 323–332. Springer, Heidelberg (2006)
12. Leest, A., Veen, M., Bruekers, F.: Reversible watermarking for images. In: Proc. of Int Conf. SPIE, Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA (January 2004)
13. Fridrich, J., Goljan, J., Du, R.: Invertible authentication. In: Proc. of Int. Conf. SPIE, Security and Watermarking of Multimedia Content, San Jose, CA, January 2001, pp. 197–208 (2001)
14. Xuan, G., Chen, J., Zhu, J., Shi, Y.Q., Ni, Z., Su, W.: Lossless data hiding based on integer wavelet transform. In: Proc. MMSP, St. Thomas, Virgin Islands, pp. 312–315 (2002)
15. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Reversible data hiding. In: Proc. IEEE ICIP, vol. 2, pp. 157–160 (2002)
16. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. on Circuits Syst. Video Technol. 13(8), 890–896 (2003)
17. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. on Image Processing 13(8), 1147–1156 (2004)
18. Lee, S., Yoo, C.D., Kalker, T.: Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform Information Forensics and Security. IEEE Trans. Info. Forensics and Security 2(3), 321–330 (2007)
19. Xuan, G.R., Shi, Y.Q., Yang, C.Y., Zheng, Y.Z., Zou, D.K., Chai, P.Q.: Lossless Data Hiding Using Integer Wavelet Transform and Threshold Embedding Technique. In: Proc. of Int. Conf. Multimedia and Expo., pp. 1520–1523 (2005)
20. Thodi, D.M., Rodriquez, J.J.: Expansion Embedding Techniques for Reversible Watermarking. IEEE Trans. Image Processing 16(3), 721–730 (2007)
21. Wu, X.: Lossless compression of continuous-tone images via context selection, quantization, and modeling. IEEE Trans. on Image Proc. 6(5), 656–664 (1997)

ASRBAC: A Security Administration Model for Mobile Autonomic Networks (MAutoNets)

Mohamad Aljnidi and Jean Leneutre

Institut Telecom, Telecom ParisTech, LTCI CNRS

Abstract. This article deals with access control in Mobile Autonomic Networks (MAutoNets), which are basically mobile ad-hoc networks. Our goal is to build an autonomic access control system. We define the Secure Relation Based Access Control (SRBAC) model, which is a variant of RBAC adapted to the MAutoNet environment by using context information and supporting autonomic computing properties. We also define the administrative counterpart of SRBAC, called ASRBAC, that allows the network nodes to manage the access control system by themselves. ASRBAC uses the distributed model ARBAC02 as a basis, and extends it with context-awareness, self-management and self-adaptation.

1 Introduction

An autonomic computing system can manage itself according to policies derived from high-level objectives [1]. It reconfigures, protects, optimizes and heals itself to adapt to changes in its environment. Its components collaborate to accomplish these tasks. Actually, an interest is growing in autonomic communications and their applications [2]. This article describes an access control model and its administrative model for autonomic mobile ad-hoc networks.

We believe that most mobile ad-hoc networks need to be autonomic. In addition to the lack of infrastructure and the variable topology, nodes are usually heterogeneous and administrators are not necessarily available. We define an evolving structure for mobile ad-hoc networks, according to which the nodes can collaborate to manage the network. We call a mobile ad-hoc network having this structure a Mobile Autonomic Network (MAutoNet). As illustrated in Figure 1, a MAutoNet is divided into communities that can be further subdivided. The nodes of a community are fix for each other with respect to the nodes of other communities. Eventually, a node may be designated as an authority for each community. For example, the set of devices of a single user may form a community, of which the user's PDA is the authority. The different authorities are then supposed to collaborate to manage the network. In a highly variable mobile ad-hoc network, each node may become a community by itself, and all nodes should cooperate as network managers. This partial centralization makes use of possible mobility limitations to reduce the self-management overhead. On the other hand, the nodes of a mobile ad-hoc network are not supposed to trust each other. However, reliability and reputation of nodes with respect to each

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 163–177, 2010.

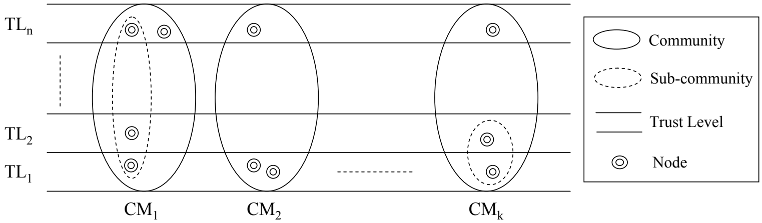


Fig. 1. MAutoNet Structure

other may differently evolve by time. As a result, nodes would be classified in trust levels, which may affect their management capabilities. Figure 1 shows that the MAutoNet structure takes the possible trust levels of nodes into account.

Generally speaking, a relation between two nodes of a mobile ad-hoc network is transient. Nevertheless, certain mobile ad-hoc networks are created to last a relatively long time, and most of their nodes are expected to keep their memberships, such as a home or a SOHO (Small Office / Home office) network. Other mobile ad-hoc networks are created for a short time, but their nodes are supposed to work together according to shared objectives, such as a mission based military network. Consequently, certain mobile ad-hoc networks need to protect their resources according to a global security model. Such complex and unexpectedly evolving networks need to be autonomic, which can be achieved by following the MAutoNet structure. In this article, we define a global access control model for MAutoNets, and its autonomic administrative model.

MAutoNet nodes would have different roles in the network with respect to their trust levels, community memberships and management capabilities. We need an access control model that assigns roles to nodes. It would be better that it supports relationships between roles for effective and scalable role management. Besides, it should support dependence upon context information (trustworthiness, mobility and authority). Moreover, because a MAutoNet is a mobile ad-hoc network, the sought access control model and its administrative model should be distributed and collaborative. Particularly, they should have self-management properties, and above all self-adaptation with respect to the evolving MAutoNet structure. A number of existing models [3,4,5,6,7,8,9] fulfill subsets of these requirements, as we explain in the next section. However, neither of them fulfills them all, and especially self-management properties.

In previous works, we described how home networks can be autonomic [10], defined the different components needed in an autonomic security system [11,12], and emphasized the need for the above access control requirements [13]. The contributions in this article consist in the definition of an access control model and its administrative counterpart fulfilling the previous requirements. The access control model called Secure Relation Based Access Control (SRBAC) is a variant of RBAC [3] adapted to the MAutoNet environment. The principal contribution of this article is the administrative model of SRBAC (ASRBAC) conceived as an extension of ARBAC02 [14]. ASRBAC allows the network nodes to manage the access control system by themselves, and it extends the distributed ARBAC02 model with context-awareness, self-management and self-adaptation.

Section 2 motivates the choice of RBAC as a basis for our access control model after discussing the related work. Section 3 presents SRBAC. Section 4 defines the Administrative model of SRBAC (ASRBAC). The Subsection 4.1 explains how certain nodes autonomously acquire administrative roles. The Subsection 4.2 describes the different types of ASRBAC administrative actions. We finally conclude by a summary of our solution and the future work in Section 5.

2 Related Work and Motivation

As explained above, we need a model that associates nodes with roles. Therefore, RBAC can be a basis for SRBAC. Particularly, RBAC supports useful hierarchical relationships between roles. Some RBAC-based approaches have been proposed for multidomain environments [4,5]. However, they imply centralization at the domain level, while a MAutoNet may be completely decentralized. On the other hand, RBAC is used as a basis in certain context-aware solutions [6,7]. However, those solutions do not support ad-hoc collaborations.

The Usage CONtrol (UCON) model [9] may also be a basis for SRBAC. It is an attribute-based model, in which persistent and mutable attributes of subjects and objects, and dynamic context-aware attributes, can be used in authorization decisions. A set of persistent and dynamic attributes can be used to define a node role in a MAutoNet. Besides, other strong concepts of UCON, such as obligations, conditions and decision continuity, provide an effective, scalable, fine-grained distributed access control. However, it is not clear how to define relationships between attributes, and administrative issues are not worked out.

The Organization Based Access Control (Or-BAC) model [8] is role-based. It supports role hierarchies, just as RBAC. It can be used in applications requiring dynamic organizational structures, such as MAutoNets, because it incorporates a context component. Besides, its administration model AdOr-BAC [15] is distributed, and suitable for collaborative systems. Moreover, the fact that AdOr-BAC is expressed using Or-BAC itself, together with the use of a context entity, may establish a framework for self-management and self-adaptation.

The Or-BAC model, and its administrative model AdOr-BAC, can be extended to support the required autonomic behavior in MAutoNets. However, a simpler enhancement of RBAC would be enough, because we currently focus on the management of roles and their associations with nodes and permissions. An extension using the advanced abstraction entities of Or-BAC, such as views and activities, will be studied in a future work. As for the UCON model, its concept of decision continuity is highly required in the ad-hoc environment of MAutoNets. Nevertheless, the main advantage of RBAC, with respect to UCON, is that it has an administrative model to build upon. Ongoing control features can be left to a future work. For these reasons we chose RBAC, and its distributed administrative model ARBAC02, as bases for the definitions of SRBAC and ASRBAC respectively. They lack context awareness, but they provide flexible role specifications where we can integrate context information. They are generic enough to be then extended with support for collaboration and autonomic computing.

3 SRBAC: The Access Control Model

Access rights are granted to a node according to its trustworthiness (trust level), availability (community membership) and management privileges with respect to the resource hosting node. When a secure relation is established between two nodes, its type is determined according to the relative values of those node attributes. Therefore, identifying the type of the secure relation determines the access permissions of the bound nodes. This is why we call the access control model of MAutoNets Secure Relation Based Access Control (SRBAC).

We present in this section the different components of SRBAC, as illustrated in Figure 2. We explain throughout this section our contribution aiming at adapting RBAC to the access control requirements of MAutoNets. Actually, SRBAC is a variant of RBAC that supports the following autonomic computing properties:

- *Context-awareness:* Trust and mobility information are integrated in node roles, and access sessions are constrained by secure relations.
- *Ad-hoc collaboration:* Nodes and objects are distributed over categories that serve as ad-hoc collaborative administration domains.
- *Self-management:* Security management privileges are taken into account in node roles, and mappings between roles and permissions are possible.

Actually, this section aims at presenting the framework of the Administrative SRBAC (ASRBAC) model, which we describe in Section 4. Therefore, SRBAC policy enforcement and implementation are out of the scope of this paper. We are currently working on the definitions of corresponding models following the layered Policy-Enforcement-Implementation (PEI) framework [16].

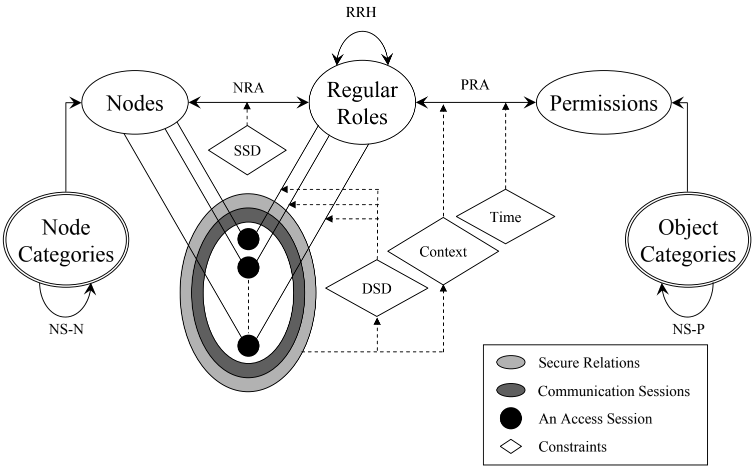


Fig. 2. The RBAC-Based MAutoNet's SRBAC Model

3.1 Node Categories

In order to have a fine-grained node classification, we need to use the trust levels and the community membership together to identify node categories. Here we define a structure of node categories based on trust and mobility.

Node trustworthiness is dynamically measured in MAutoNets with respect to evolving reputation criteria. We assume that a reputation system is already implemented in a MAutoNet. Some existing reputation systems provide the dynamic measure of trustworthiness required in ad-hoc applications [17,18]. We currently assume that the reputation system assigns each node to a trust level that has a global scope in the network. There are initially two trust levels in a MAutoNet: H (*Highest*) and L (*Lowest*). The reputation system may change the trust levels of nodes, and possibly add new trust levels.

Definition 1. *Let $\mathcal{T}$ be the set of trust levels, $\mathcal{N}$ the set of nodes and $tLevel$ the function which returns the trust level of a node:*

$$tLevel : \mathcal{N} \longrightarrow \mathcal{T}$$

Each MAutoNet node belongs to a community, but it may also belong to a sub-community. Community subdivision loosen mobility constraints, which is required in ad-hoc environments. In order to reduce complexity in our current work, we support only one-depth community subdivision.

Definition 2. *Let $\mathcal{C}$ be the set of communities and $nComm$ the function which returns the community of a node:*

$$\mathcal{C} \subseteq 2^{\mathcal{N}}$$
$$nComm : \mathcal{N} \longrightarrow \mathcal{C}$$
$$\forall x \in \mathcal{N}, (\exists c \in \mathcal{C}, nComm(x) \subseteq c) \Rightarrow (x \in c)$$

Definition 3. *Let NC be the set of node categories and $nPool$ the function which returns the set of nodes assigned the same node category:*

$$NC \subseteq (\mathcal{T} \cup \{\mathcal{N}\}) \times \mathcal{C}$$
$$nPool : NC \longrightarrow 2^{\mathcal{N}}$$
$$nPool(\mathcal{N}, \mathcal{N}) = \mathcal{N}$$
$$\forall t \in \mathcal{T}, nPool(t, \mathcal{N}) = \{x \in \mathcal{N} \mid tLevel(x) = t\}$$
$$\forall t \in \mathcal{T}, \forall c \in \mathcal{C}, nPool(t, c) = \{x \in \mathcal{N} \mid (tLevel(x) = t) \wedge (nComm(x) = c)\}$$

Definition 4. *Let X and Y be two node categories, we write $X \succeq_{NC} Y$, and we say that X dominates Y in NC, if and only if the nodes of Y belong to X:*

$$\forall X, Y \in NC, (X \succeq_{NC} Y) \Leftrightarrow (nPool(Y) \subseteq nPool(X))$$

The operation $\succeq_{NC}$ defines a partial order creating a tree structure, in which a category inherits nodes from its descendants. We call this tree NS-N (Network Structure for Nodes). Figure 3 illustrates the NS-N of a MAutoNet having two communities $c1$ and $c2$, and the default set of trust levels. The direction of the arrows represents the origin of a node membership.

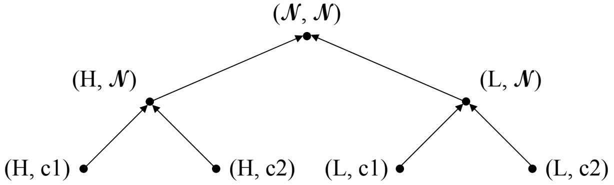


Fig. 3. Example of a Network Structure for Nodes (NS-N)

3.2 Object Categories

In a MAutoNet, certain attributes of a node identify the category of objects that it can access. We already defined the node attributes called trust level and community membership. A third node attribute, called basic role, is needed to identify the category of potential target objects. It defines management capabilities for a node. Basic roles follow a total order representing an administrative privilege increase. A node designated as the authority of one or more communities may have the highest basic role A in certain secure relations. The lowest basic role n is assigned to nodes that have no rights in terms of security management.

Definition 5. *Let SR be the set of secure relations, $\mathcal{B}$ the set of basic roles and bRole the function which returns the basic role of a node in a secure relation:*

$bRole : \mathcal{N} \times SR \longrightarrow \mathcal{B}$

Actually, an object is assigned to one or more of a set of attributes representing authorization scopes, with respect to the trust levels, community memberships and/or basic roles of the nodes requesting the access. On the other hand, a certain composition of authorization scopes defines a set of permissions. This is actually why the attributes of a node, as evaluated in a secure relation, define its permissions. The following list describes the possible authorization scopes:

1. $\alpha_{\mathcal{T},t}$: Access is unauthorized at trust levels lower than t.
2. $\alpha_{\mathcal{C},c}$: Access is restricted to the community c.
3. $\alpha_{\mathcal{B},b}$: Access is unauthorized for basic roles lower than b.

Definition 6. *Let $\mathcal{P}$ be the set of permissions, Att_p the set of authorization scopes associated with a permission p, OC the set of object categories and pPool the function which returns the set of permissions assigned to an object category:*

$OC \subseteq (\mathcal{T} \cup \{\mathcal{N}\}) \times \mathcal{C} \times (\mathcal{B} \cup \{\mathcal{N}\})$
$pPool : OC \longrightarrow 2^{\mathcal{P}}$
$pPool(\mathcal{N}, \mathcal{N}, \mathcal{N}) = \mathcal{P}$
$\forall t \in \mathcal{T}, pPool(t, \mathcal{N}, \mathcal{N}) = \{p \in \mathcal{P} \mid \alpha_{\mathcal{T},t} \in Att_p\}$
$\forall t \in \mathcal{T}, \forall c \in \mathcal{C}, pPool(t, c, \mathcal{N}) = \{p \in \mathcal{P} \mid \{\alpha_{\mathcal{T},t}, \alpha_{\mathcal{C},c}\} \subseteq Att_p\}$
$\forall t \in \mathcal{T}, \forall c \in \mathcal{C}, \forall b \in \mathcal{B}, pPool(t, c, b) = \{p \in \mathcal{P} \mid \{\alpha_{\mathcal{T},t}, \alpha_{\mathcal{C},c}, \alpha_{\mathcal{B},b}\} \subseteq Att_p\}$

Definition 7. *Let X and Y be two object categories, we write $X \succeq_{OC} Y$, and we say that X dominates Y in OC, if and only if the permissions assigned to X are assigned to Y:*

$$\forall X, Y \in OC, (X \succeq_{OC} Y) \Leftrightarrow (pPool(X) \subseteq pPool(Y))$$

The operation $\succeq_{OC}$ defines a partial order creating a tree structure, in which a category inherits permissions from its ascendants. We call this tree NS-P (Network Structure for Permissions). Figure 4 illustrates the NS-P of a MAutoNet having two communities $c1$ and $c2$, and the default sets of trust levels and basic roles. The direction of the arrows represents the origin of a permission.

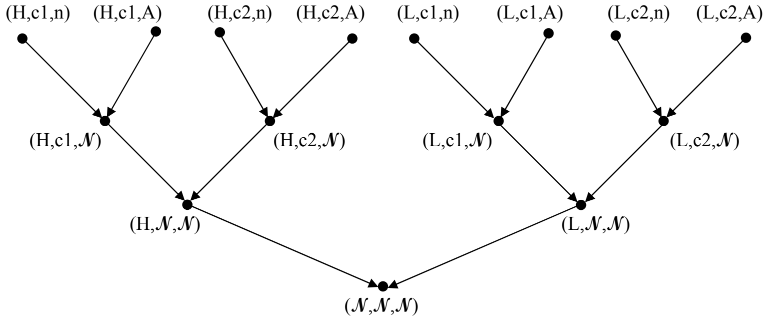


Fig. 4. Example of a Network Structure for Permissions (NS-P)

3.3 Regular Roles

The actual role of a node, called *Regular Role*, is characterized by its trust level, community membership and basic role altogether. In case of a node having administrative privileges, the regular role is also characterized by the set of communities it manages.

Definition 8. *Let $aComm$ be the function which returns the set of communities managed by a node with regard to a given basic role:*

$$aComm : \mathcal{N} \times \mathcal{B} \longrightarrow 2^{\mathcal{C}}$$

Definition 9. *Let RR be the set of regular roles and $rRole$ the function which returns the regular role of a node in a secure relation:*

$$RR \subseteq \mathcal{T} \times \mathcal{C} \times \mathcal{B} \times 2^{\mathcal{C}}$$
$$rRole : \mathcal{N} \times SR \longrightarrow RR$$
$$\forall x \in \mathcal{N}, \forall \rho \in SR, \exists b \in \mathcal{B}, bRole(x, \rho) = b,$$
$$\quad rRole(x, \rho) = (tLevel(x), nComm(x), b, aComm(x, b))$$

Definition 10. *Let S_1 and S_2 be two sets of communities, we write $S_1 \succeq S_2$, and we say that S_1 dominates S_2, if and only if the nodes of any community in S_2 belong to a community in S_1:*

$$\forall S_1, S_2 \in 2^{\mathcal{C}}, (S_1 \succeq S_2) \Leftrightarrow (\forall s_2 \in S_2, \exists s_1 \in S_1, s_2 \subseteq s_1)$$

Definition 11. *Let X and Y be two regular roles, we write $X \succeq_{RR} Y$, and we say that X dominates Y in RR, if and only if the trust level in X is higher than the trust level in Y, the community in X is a part of the community in Y, the basic role in X is higher than the basic role in Y, and the set of controlled communities in X dominates the set of controlled communities in Y:*

$$\forall X, Y \in RR, X = (t1, c1, b1, s1), Y = (t2, c2, b2, s2),$$
$$(X \succeq_{RR} Y) \Leftrightarrow ((t1 \geq t2) \wedge (c1 \subseteq c2) \wedge (b1 \geq b2) \wedge (s1 \succeq s2))$$

The operation $\succeq_{RR}$ defines a partial order creating a lattice hierarchy, in which a regular role inherits privileges from its descendants. This is the hierarchy that we call RRH (Regular Role Hierarchy). Figure 5 illustrates a part of the RRH of a MAutoNet having two communities $c1$ and $c2$, and the default sets of trust levels and basic roles. The direction of the arrows indicates the propagation of privileges from a role to its senior roles by inheritance.

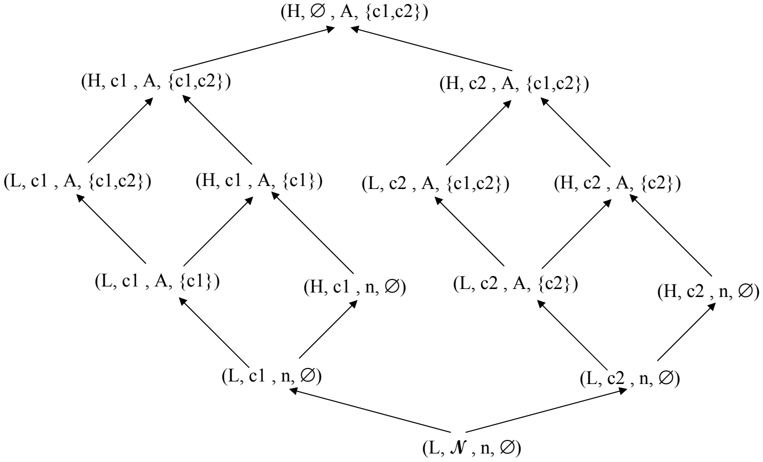


Fig. 5. Example of a Regular Role Hierarchy (RRH)

3.4 Node-Role Assignment (NRA)

Each MAutoNet node must have one role at least. A new node x would initially have either the role $(tLevel(x), comm(x), A, \{comm(x)\})$ if it is configured to be the authority of its community, or the role $(tLevel(x), comm(x), n, \emptyset)$ otherwise. Eventually, other roles would be assigned to the new node according to the inheritance relations in RRH. Besides, roles may be assigned to a node irrespectively of inheritance, due to the network evolution. On the other hand, many nodes can be assigned to one role. For instance, all the members of a node category $(x, y) \in NC$ are assigned to the role $(x, y, n, \emptyset)$, either directly or by inheritance. NRA is a many-to-many relationship as illustrated in Figure 2.

3.5 Permission-Role Assignment (PRA)

Basically, a permission is a relationship between an action and a target object. Certain models enhance this representation, such as OrBAC [8] that provides a flexible abstraction of actions and objects using activities and views respectively. Such abstraction would be useful in SRBAC, where we deal with categories of objects. However, we currently do not work on the concrete format of a permission. We just care about the categorization of permissions with respect to object categories, which is needed for managing permission-role assignments. PRA is based on a mapping between regular roles and object categories on one hand, and between object categories and permission sets on the other hand. Such mappings allow the administrative model ASRBAC to apply self-configuration mechanisms. Just as in RBAC, PRA is a many-to-many relationship in SRBAC, as illustrated in Figure 2.

Definition 12. *We define the function rObjects which returns the set of object categories assigned to a role by mapping.*

$$\begin{aligned}
&rObjects : RR \longrightarrow 2^{OC}\\
&\forall r \in RR : (\exists t \in \mathcal{T}, \exists c \in \mathcal{C} : r = (t, c, n, \emptyset)) \Leftrightarrow (rObjects(r) = \{(t, c, n)\})\\
&\forall r \in RR : (\exists t \in \mathcal{T}, \exists c \in \mathcal{C}, \exists b \in \mathcal{B} : b \neq n, \exists s \in 2^{\mathcal{C}}, r = (t, c, b, s))\\
&\qquad \Leftrightarrow (rObjects(r) = \{(t, d, b) \in OC \mid d \in s\})
\end{aligned}$$

3.6 Access Sessions

A subject in RBAC creates an access session by activating one or more of its roles to perform actions on certain objects. In SRBAC, a secure relation encapsulates each access session, and only the regular role of the access requesting node in that secure relation is activated. However, a node may activate many regular roles in parallel access sessions to simultaneously access objects on different nodes.

3.7 Constraints

Certain constraints in SRBAC are dynamic and depend on the secure relation, such as the DSD (Dynamic Separation of Duty) constraints applied on parallel session-role assignments, and the *Context* constraints. Others are either static, such as the SSD (Static Separation of Duty) constraints, or dynamic but independent of the secure relation, such as the *Time* constraints. We particularly need context and time constraints to fulfill the requirements of the dependence upon secure relations and the ad-hoc evolution respectively.

4 ASRBAC: The Administration Model

The SRBAC components NS-N, NRA, RRH, PRA and NS-P, illustrated in figure 2, are initially derived from the high-level configuration done by end-users at the MAutoNet deployment time. Afterward, these components must adapt

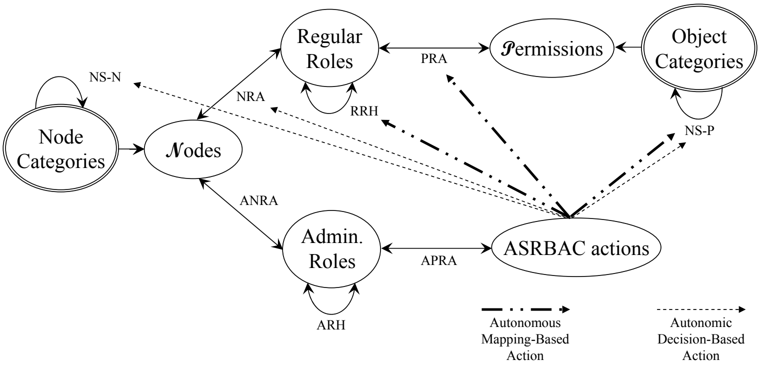


Fig. 6. Components of ASRBAC and its relation with SRBAC

to the ad-hoc evolution of the MAutoNet. This adaptation must be achieved using autonomic mechanisms, by conformity to the nature of MAutoNets. Actually, we define a model for autonomic Administration of SRBAC, and we call it ASRBAC.

ASRBAC is an extended version of ARBAC02 [14] adding autonomic behavior. ASRBAC makes use of the support of distributed administration in ARBAC02, which is required in mobile ad-hoc environments, and extends it with self-management, which is required in MAutoNets.

As illustrated in figure 6, a node may be simultaneously assigned to a set of regular roles in the context of SRBAC, and to a set of administrative roles in the context of ASRBAC. More specifically, a node having the authority basic role A autonomously acquires a corresponding administrative role. In other words, the Administrative Node-Role Assignment (ANRA) relationships are spontaneous, and ARH is the result of a mapping to RRH (see figure 8). On the other hand, figure 6 shows that administrative permissions are not selected from the permission pools of NS-P. Actually, the Administrative Permission-Role Assignment (APRA) relationships are autonomously defined according to the correspondence between the attributes of an administrative role and the attributes of the SRBAC components it may control.

4.1 Administrative Roles

For specifying the administrative role of an authority node, we just need to know its trust level and the set of controlled communities. Its community membership does not affect its administrative privileges, and its basic role is A by definition.

Definition 13. *We define the set of administrative roles AR, and the function aRoles which returns the set of administrative roles of a node.*

$AR \subseteq \mathcal{T} \times 2^{\mathcal{C}}$
$aRoles : \mathcal{N} \longrightarrow 2^{AR}$
$\forall x \in \mathcal{N} : aRoles(x) = \{(t, s) \mid \exists r \in nRoles(x), \exists c \in \mathcal{C} : r = (t, c, A, s)\}$

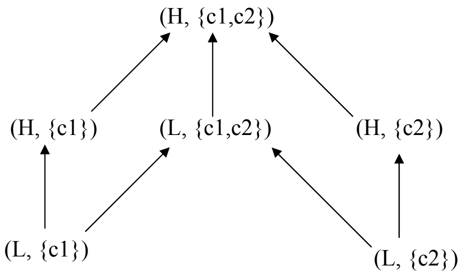


Fig. 7. Example of an Administrative Role Hierarchy (ARH)

Definition 14. *Given the two administrative roles X and Y, we write $(X \succeq_{AR} Y)$, and we say that X dominates Y in AR, if and only if the trust level in X is higher than the trust level in Y and the set of controlled communities in X dominates the set of controlled communities in Y.*

$$\forall X, Y \in AR : X = (t1, s1), Y = (t2, s2), (X \succeq_{AR} Y) \Leftrightarrow ((t1 \geq t2) \wedge (s1 \succeq s2))$$

The operation $\succeq_{AR}$ defines a partial order creating a hierarchy, in which an administrative role inherits privileges from its descendants. We call this hierarchy ARH (Administrative Role Hierarchy). Figure 7 illustrates the ARH of a MAutoNet having two communities $c1$ and $c2$, and the default set of trust levels.

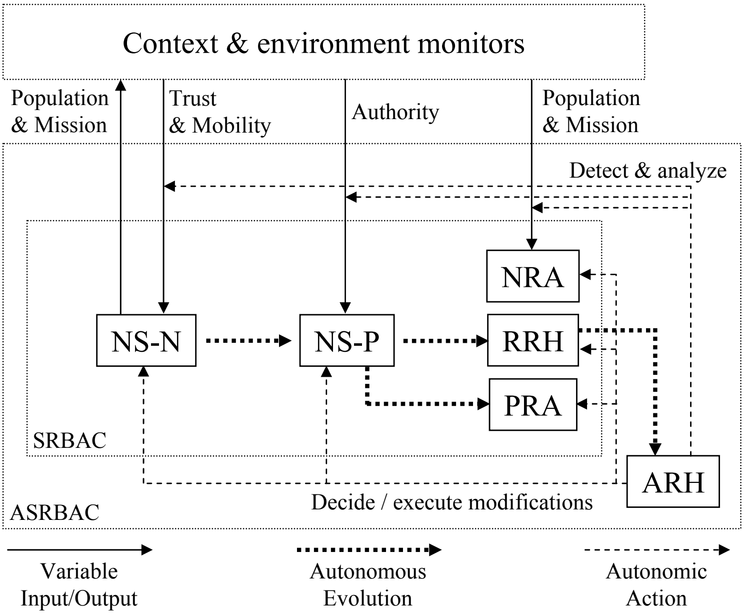


Fig. 8. Administrative Actions in ASRBAC, and Application in MAutoNets' Context

4.2 Autonomic Behavior

As illustrated in Figure 8, the authority nodes need to detect and analyze context information, in order to perform administrative actions. A set of context-aware systems provides them with the required environment variables. A reputation system may output the environment variables *Trust* and *Authority* describing changes in trust levels and basic roles respectively, and a mobility management system may output the environment variables *Mobility*, *Population* and *Mission* describing changes in community composition, node membership and node roles respectively. The reputation system and the mobility system are therefore key elements, but their description is out of the scope of this paper.

Predefined Self-Management. Certain decision-based administrative actions in ASRBAC apply a predefined self-management functionality. Such administrative actions occur in response to expected changes. The environment variables reporting expected changes are considered as non-critical because they do not lead to changes in the autonomic system itself [19]. This is the case of the administrative actions applied on NRA in ASRBAC, because they do not change ASRBAC components. The non-critical variables that would produce changes to NRA in MAutoNets are *Population* and *Mission*. We can see in figure 8 that changes in NRA do not imply further changes in ARH.

Example 1. Due to the ad-hoc nature of a MAutoNet, it is expected that an authority node *auth* leaves the network. The environment variable *Population* reports this event to the other authority nodes of the network. If one of them has an administrative role senior to $aRoles(auth)$ it removes all the node-role assignments corresponding to *auth* from NRA, otherwise they negotiate and agree to perform this deletion together. Besides, one or more communities would have no authority node as a result, which is detected by the authority nodes through the environment variable *Mission*. In response to this event, the authority nodes negotiate to select a qualified node to assign the required authority role to it.

Autonomic Control Loop. An autonomic system should be able to cope with unexpected changes in its environment by adapting itself [1]. Such functionality is achieved through autonomic control loops [2], whereby the execution of the autonomic actions result in a feedback that may change the autonomic system itself. Unexpected changes are detected through monitoring critical, essential, environment variables [19], such as *Trust*, *Mobility* and *Authority* in the case of ASRBAC. Such changes make the authority nodes decide to modify SRBAC components, which is basically the goal of administrative actions, but they will also make ASRBAC components adapt accordingly. Figure 8 illustrates the autonomic control loops of ASRBAC. We can see that a change in NS-N or in NS-P ends by a change in RRH, which in its turn implies changes in ARH and eventually in the administrative role assignments ANRA and APRA.

Example 2. A reputation system may decide that a new trust level t_{new} must be added. The authority nodes capture this critical decision through detection and analysis of changes in the environment variable *Trust*. The following modifications then take place (see figure 8):

1. *Decision-based update of NS-N:* Creation of a new node category $(t_{new}, \mathcal{N})$, and the set of its junior node categories $\{(t_{new}, c) \mid c \in \mathcal{C}\}$. Certain nodes are then removed from their respective node categories to be added to the new ones. NS-N will be then updated by integrating the new node categories.
2. *Mapping of NS-P:* The node category $(t_{new}, \mathcal{N})$ implies the creation of the new object category $(t_{new}, \mathcal{N}, \mathcal{N})$, and the set of node categories $\{(t_{new}, c) \mid c \in \mathcal{C}\}$ implies the creation of the sets of object categories $\{(t_{new}, c, \mathcal{N}) \mid c \in \mathcal{C}\}$ and $\{(t_{new}, c, b) \mid c \in \mathcal{C}, b \in \mathcal{B}\}$. NS-P will be then updated by integrating the new object categories.
3. *Mapping of RRH:* For each element in the new set of object categories $\{(t_{new}, c, b) \mid c \in \mathcal{C}, b \in \mathcal{B}\}$, if $b = n$ then the regular role $(t_{new}, c, b, \emptyset)$ is created, otherwise the set of regular roles $\{(t_{new}, c, b, s) \mid s \in 2^{\mathcal{C}} - \{\emptyset, \mathcal{N}\}\}$ is created. RRH will be then updated by integrating the new regular roles.
4. *Spontaneous update of ARH:* The set of regular roles $\{(t_{new}, c, b, s) \mid c \in \mathcal{C}, b = A, s \in 2^{\mathcal{C}} - \{\emptyset, \mathcal{N}\}\}$, which indicates potential new assignments to authority nodes in the network, causes the new set of administrative roles $\{(t_{new}, s) \mid s \in 2^{\mathcal{C}} - \{\emptyset, \mathcal{N}\}\}$ to be created and integrated in ARH.

The administrative actions performed in the context of an autonomic control loop in ASRBAC may imply further administrative actions of different types. Figure 8 shows that changes to NS-N may cause changes to the values of the non-critical environment variables *Population* and *Mission*, which will lead to performing administrative actions on NRA as explained previously. For instance, changes in the mobility conditions of the network, captured through the environment variable *Mobility*, may make authority nodes decide to merge two communities. This necessarily implies the modification of NS-N because of the revocation of the two merged communities and the integration of the new community resulting from this administrative action. Nevertheless, this modification is achieved through intermediary node removals and insertions, which are captured as changes to the environment variable *Population*. This may possibly lead to the resignation of certain authority nodes and the assignation of others, which implies changes to the environment variable *Mission*.

Autonomous Evolution. In addition to autonomic decision-based administrative actions, ASRBAC defines autonomous mapping-based administrative actions[1] as illustrated in figure 6. In such cases, authority nodes cooperate just to realize a mapping between two components of SRBAC in response to changes in one of them. Figure 8 illustrates the following autonomous evolution kinds:

1. A change to NS-N implies autonomous evolution of NS-P, as clarified above in step 2 of Example 2. A mapping is possible because NS-P is the inverse of NS-N extended by basic roles.

[1] An autonomic action involves a collaborative analysis of context information and decision making, while an autonomous action executes mappings right after detecting changes without any analysis or decision efforts.

2. A change to NS-P implies autonomous evolution of RRH, as clarified above in step 3 of Example 2. A mapping is possible according to the definition of the function *rObjects* (Definition 12).
3. NS-P changes are also behind the autonomous evolution of PRA. More precisely, each time NS-P changes, the authority roles review the high-level configuration of node privileges and redistribute derived permissions on object categories; hence on corresponding regular roles. This is also done each time the high-level configuration is updated, which is captured by a dedicated context-aware system and reproduced as a change to the essential environment variable *Authority*. Eventually, this makes the authority nodes decide to change NS-P, and consequently PRA.

5 Conclusion

We introduced a network structure that helps building Mobile Autonomic Networks (MAutoNets) out of mobile ad-hoc networks. We defined the Secure-Relation-Based Access Control (SRBAC) model of MAutoNets, which is based on the well-recognized RBAC model [3]. It makes use of the flexibility of RBAC and adapts it to the MAutoNet requirements. The main contribution of this article was the autonomic Administrative SRBAC (ASRBAC) model. We described the different kinds of administrative actions of ASRBAC, which allow the access control system of a MAutoNet to be decentralized, collaborative, self-managing and self-adapting. We currently work out SRBAC policy enforcement. Afterward, we will study how authority nodes negotiate SRBAC policies in their cooperative enforcement of ASRBAC policies.

Our solution considered role management issues only. We still need autonomic solutions for the management of access sessions, constraints and permission specification. In a future work, we will extend SRBAC with the components of Or-BAC [8] to achieve this goal. Another limitation of our solution is the lack of a support for ongoing access control during an access session, which is required in ad-hoc applications. The UCON model [9] provides such capabilities through mutable attributes and decision continuity. A future work could be to express SRBAC, or its future version built over Or-BAC, using the generic UCON policies, and then propose an accompanying autonomic administrative model.

References

1. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer (2003)
2. Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (2006)
3. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer (1996)
4. Shafiq, B., Joshi, J., Bertino, E., Ghafoor, A.: Secure interoperation in a multidomain environment employing rbac policies. IEEE Trans. Knowl. Data Eng. (2005)

5. Shehab, M., Bertino, E., Ghafoor, A.: Secure collaboration in mediator-free environments. In: Proceedings of the 12th ACM Conference on Computer and Communication Security (2005)
6. Covington, M.J., Long, W., Srinivasan, S., Dey, A.K., Ahamad, M., Abowd, G.D.: Securing context-aware applications using environment roles. In: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies (2001)
7. Zhang, G., Parashar, M.: Dynamic context-aware access control for grid applications. In: Proceedings of the 4th International Workshop on Grid Computing (2003)
8. Abou El Kalam, A., Baida, R.E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miege, A., Saurel, C., Trouessin, G.: Organization based access control. In: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (2003)
9. Zhang, X., Parisi-Presicci, F., Sandhu, R.: Formal model and policy specification of usage control. ACM Transactions on Information and System Security (2005)
10. Aljnidi, M., Leneutre, J.: Autonomic security for home networks. In: Proceedings of the First International Workshop on Self-Organizing Systems (2006)
11. Aljnidi, M., Leneutre, J.: Towards an autonomic security system for mobile ad hoc networks. In: Proceedings of the Third International Symposium on Information Assurance and Security (2007)
12. Aljnidi, M., Leneutre, J.: A security policy system for mobile autonomic networks. In: Proceedings of the First International Conference on Autonomic Computing and Communication Systems (2007)
13. Aljnidi, M., Leneutre, J.: Security solutions in mobile autonomic networks. In: Proceedings of the Third International Conference on Information and Communication Technologies: From Theory to Applications (2008)
14. Oh, S., Sandhu, R., Zhang, X.: An effective role administration model using organization structure. ACM Transactions on Information and System Security (2006)
15. Cuppens, F., Miege, A.: Administration model for or-BAC. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2003. LNCS, vol. 2889, pp. 754–768. Springer, Heidelberg (2003)
16. Sandhu, R., Ranganathan, K., Zhang, X.: Secure information sharing enabled by trusted computing and pei models. In: Proceedings of the ACM Symposium on Information, Computer and Communication Security (2006)
17. Pirzada, A.A., McDonald, C.: Trust establishment in pure ad-hoc networks. Wireless Personal Communications (2006)
18. Michiardi, P., Molva, R.: Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Proceedings of the IFIP 6th Joint Working Conference on Communications and Multimedia Security (2002)
19. Hariri, S., Khargharia, B., Chen, H., Yang, J., Zhang, Y., Parashar, M., Liu, H.: The autonomic computing paradigm. Cluster Computing (2006)

Untraceable Tags Based on Mild Assumptions*

Carlo Blundo, Angelo De Caro, and Giuseppe Persiano

Dipartimento di Informatica ed Applicazioni, Università di Salerno, Italy

Abstract. Radio frequency identification (RFID) chips have been widely deployed in large-scale systems such as inventory control and supply chain management. While RFID technology has much advantage, however it may create new problems to privacy. Tag untraceability is a significant concern that needs to be addressed in deploying RFID-based system.

In this paper we propose a new construction for untraceable tags. Our construction is the first construction in the symmetric bilinear setting based on a *mild* assumption. That is our assumption is tautological in the generic group model and is "efficiently falsifiable" in the sense that its problem instances are stated non-interactively and concisely (i.e., independently of the number of adversarial queries and other large quantities).

1 Introduction

RFID [5] and NFC [10] are the *de-facto* technology for storing small amount of data on devices that can be read without physical contact. It is expected that everyday objects will be tagged with small components which are used to carry information to identify the object. For example, the garnment industry plans to use RFID tags for the management of post-sale services. Obviously, it is expected that encryption is used for storing information on the tag so that only legitimate users can access the stored data. Encryption though does not solve all problems and we are interested in privacy issues associated with RFID tags. Specifically, RFID tags can be read by anyone and the string stored on a tag, even though it is a ciphertext, can be used to trace the tag and, in the case the tag is attached to a personal object, to trace the owner of the tag.

We thus envision a system in which the environment helps in alleviating this problem: as tags move in the environment they are read by special devices called the *randomizers* which provide the following service: everytime a randomizer reads a tag carrying a ciphertext, the ciphertext is re-randomized; that is, a new ciphertext carrying the same cleartext is computed. This can be easily achieved if the randomizers are trusted with the secret keys: just decrypt the ciphertext to obtain the cleartext and then encrypt the cleartext using fresh randomness. In some applications though this is a very strong trust assumption: even if one of the randomizers is corrupted then all privacy is lost. We thus look at the

* The work of the authors has been supported in part by the European Commission through the FP7 Information Communication Technologies programme, under Contract FET-215270 FRONTS (Foundations of Adaptive Networked Societies of Tiny Artefacts).

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 178–192, 2010.

problem of designing special encryption schemes that support re-randomization; that is, given a ciphertext Ct carrying cleartext M, it is possible to produce a new ciphertext Ct' carrying the same cleartext M, even if the decryption key is not available.

The El-Gamal encryption scheme. A simple variation of the El-Gamal encryption scheme is known to be re-randomizable [6], but it is of limited applicability. Let us review the re-randomizable version of the ElGamal encryption scheme.

Let p be a large prime and let g be a generator of $\mathbb{Z}_p^\star$. The public key for the ElGamal encryption scheme consists simply of an element $y \in \mathbb{Z}_p^\star$ and the associated secret key is $x \in \mathbb{Z}_p^\star$ such that $y = g^x$ (all operations are in $\mathbb{Z}_p^\star$). In the encryption scheme $\mathsf{rElGamal}$ (the re-randomizable version of the ElGamal encryption scheme), to encrypt message $M \in \mathbb{Z}_p^\star$, one selects $r, s \in Z_p^\star$ at random and computes the pair (g^r, My^r, g^s, y^s). The plaintext associated to ciphertext $\mathsf{Ct} = (C_0, C_1, U_0, U_1)$ is recovered by computing C_1/C_0^x, where x is the secret key. The re-randomization procedure takes a ciphertext $\mathsf{Ct} = (C_0, C_1, U_0, U_1)$, selects $t, t' \in Z_p^\star$ at random and returns $\widehat{\mathsf{Ct}} = (C_0 \cdot U_0^t, C_1 \cdot U_1^t, U_0^{t'}, U_1^{t'})$. It is easy to see that if Ct is a ciphertext for cleartext M then $\widehat{\mathsf{Ct}}$ is a uniformly distributed ciphertext for the same cleartext M. Also notice that the re-randomization procedure does not need to know neither the public key nor the secret key associated with the ciphertext Ct.

Suppose now that we want to store message M on a tag and suppose we use $\mathsf{rElGamal}$ to encrypt M before actually storing on the tag. Unfortunately, an adversary $\mathcal{A}$ that wants to trace a tag has a very simple and successful strategy. $\mathcal{A}$ simply generates a pair of public/secret key $(y_\mathcal{A}, x_\mathcal{A})$ for $\mathsf{rElGamal}$ and writes a random message $M_\mathcal{A}$ on the tag $T_\mathcal{A}$ that he wants to trace by computing ciphertext $\mathsf{Ct}_\mathcal{A}$ for public key $y_\mathcal{A}$. Notice that everytime $T_\mathcal{A}$ is re-randomized by the randomizers, message $M_\mathcal{A}$ is not affected. Thus to check that a given tag T is actually $T_\mathcal{A}$, $\mathcal{A}$ can simply try to decrypt the stored ciphertext and if the decryption gives back $M_\mathcal{A}$ then with very high probability $\mathcal{A}$ can conclude that he is in presence of $T_\mathcal{A}$.

We notice that $\mathsf{rElGamal}$ can still be used in the scenario in which writing on the tag can be selectively disabled by the owner. That is, the owner of the tag enables writing on the tag when in presence of trusted randomizers and disables writing if he is in an untrusted environment.

The scenario. In this paper, we consider the more challenging scenario in which writing on a tag cannot be selectively disabled. Obviously, in this scenario, an adversary $\mathcal{A}$ can destroy the content of a tag T by overwriting its content. We will guarantee though that $\mathcal{A}$ cannot trace tag T even in this case.

We have three types of honest players:

1. The Central Authority CA that publishes some public information Pub and issues a pair of private and secret keys to each authorized player.
2. The players that receive a public and secret key from the CA and use the keys to encrypt and decrypt messages that are stored on tags.

3. The randomizers that receive tags and randomize the ciphertexts stored on the tags. The randomization procedure changes the ciphertext but not the cleartext stored on the tag.

Notice that the role of the CA is necessary: if users could generate keys by themselves then the it would not be possible to prevent attacks similar to the one we have discussed for the $\mathsf{rElGamal}$ encryption scheme.

In this paper we give a construction for untraceable tags. We split the presentation in two parts. In Section 4 we present a tag system that is secure against adversaries that can only read tags. Building on the construction of Section 4, in Section 5 we present out main result, a tag system that is secure against adversaries that can write on tags.

Previous work. In [1], a construction for an untraceable tag system was proposed. The security of the construction of [1] is based on a stronger version of the LRSW assumption introduced by Lysyanskaya et al. [7]. The strong LRSW assumption does not hold for symmetric bilinear mapping. Specifically, the construction of [1] requires the existence of three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ such that no morphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ exists and of a bilinear mapping $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. This is called the *asymmetric* bilinear setting. If one tries to use the construction of [1] in the symmetric bilinear setting then, as it is easily seen, tags become traceable. Our construction instead is in the *symmetric* bilinear setting. In [1] the authors state that in the full version of their paper they will show a construction of the symmetric bilinear setting. To the best of our knowledge such a full version was never published.

Moreover, our construction is based on a *mild* assumption in the sense of [3]. That is our assumption is tautological in the generic group model [12] and is "efficiently falsifiable" [9] in the sense that its problem instances are stated non-interactively and concisely (i.e., independently of the number of adversarial queries and other large quantities). In contrast, the assumption used to prove the security of the construction in [1] is stated in an interactive way.

2 The Model

We start by defining the notion of a tag system and then define its security properties. We consider quintuples of algorithms (GenPub, GenKey, rEnc, rDec, $\mathsf{Randomize}$) with the following intended meaning.

1. $\mathsf{GenPub}(1^k)$ is executed by the CA. It takes as input the security parameter k and returns the public information Pub, the randomizing information rPub, and the master secret key Msk.
2. $\mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$ is executed by the CA to generate the secret key of a player. It takes as input the public information Pub and the master secret key Msk and returns the public key Pk and the secret key Sk.
3. $\mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ is executed by a player to encrypt a message M to be written on a tag. It takes as input the public information Pub, the public key Pk of

the user for which the message is encrypted, and the message M and returns the ciphertext Ct.

4. $\mathsf{rDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct})$ is executed by a player to decrypt a ciphertext Ct. It takes as input the public information Pub, the secret key Sk of the user, and the cipheretext Ct and returns the cleartext M.
5. $\mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct})$ is executed by the randomizers to randomize ciphertexts. It takes as input the public information Pub, the randomizing information rPub, and a ciphertext Ct that encrypts a message M for public key Pk and returns a new ciphertext $\mathsf{Ct}^\star$ that encrypts message M for Pk. We stress that M, Pk, and the secret key Sk are not given as input to $\mathsf{Randomize}$.

In a typical scenario, the players are manufacturers that attach tags to consumer goods. They obtain their pair of private and secret key from the CA and use the encryption algorithm rEnc to store information regarding the good on the tag. We envision randomizers being present in the physical environment were the end user lives. Finally, the decryption algorithm rDec is used by the manufacturer to recover the information written on the tag when the end user requires assistance (or maintenance) from the manufacturer.

Definition 1. *A* tag system *is a quintuple of algorithms (*GenPub*,* GenKey*,* rEnc*,* rDec*,* Randomize*) such that for any* $\ell = \mathsf{poly}(k)$*,*

$$\begin{aligned}\text{Prob}[&(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k); (\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk});\\ &\mathsf{Ct}_0 \leftarrow \mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M});\\ &\mathsf{Ct}_1 \leftarrow \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct}_0);\\ &\ldots;\\ &\mathsf{Ct}_\ell \leftarrow \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct}_{\ell-1}); \mathsf{M}' \leftarrow \mathsf{rDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct}_l) : \mathsf{M} = \mathsf{M}'] = 1\end{aligned}$$

We next define the security properties of a tag system. We start from semantic security.

Semantic security. Consider the following experiment with an adversary $\mathcal{A}$.

$\mathsf{SSExp}_{\mathcal{A}}(1^k)$

1. $(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k)$.
2. $(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$.
3. Run $\mathcal{A}$ on input Pub and Pk and obtain messages $\mathsf{M}_0, \mathsf{M}_1$.
4. Toss a random coin $\eta \in \{0, 1\}$ and compute $\mathsf{Ct} = \mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M}_\eta)$.
5. Run $\mathcal{A}$ on input Ct and let η' be its output.
6. If $\eta = \eta'$ then return 1 else return 0.

In $\mathsf{SSExp}_{\mathcal{A}}$ the adversary $\mathcal{A}$ selects two strings of his choice, M_0 and M_1. Then, one of the strings is picked at random, it is encrypted and given to the adversary. We require that the adversary is not able to guess which of the two string has been encrypted.

Definition 2. *A tag system (*GenPub*,* GenKey*,* rEnc*,* rDec*,* Randomize*) is* semantically secure *if for all probabilistic polynomial-time algorithms* $\mathcal{A}$ *we have that*

$$\left|\text{Prob}[\,\mathsf{SSExp}_{\mathcal{A}}(1^k) = 1\,] - \frac{1}{2}\right|$$

is negligible in k.

Weak untraceability. Next we define the notions of untraceability for a tag systems. We start with the notion of weak untraceability and then present our notion of strong untraceability. For defining the notion of weak untraceability we use the following experiment.

```
$\mathsf{WUExp}_{\mathcal{A}}(1^k)$
 1. $(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k)$.
 2. $(\mathsf{Pk}_0, \mathsf{Sk}_0) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$ and $(\mathsf{Pk}_1, \mathsf{Sk}_1) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$.
 3. Run $\mathcal{A}$ on input $\mathsf{Pub}, \mathsf{Pk}_0$ and $\mathsf{Pk}_1$ and obtain messages $\mathsf{M}_0, \mathsf{M}_1$.
 4. Compute $\mathsf{Ct}_0 = \mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}_0, \mathsf{M}_0)$, $\mathsf{Ct}_1 = \mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}_1, \mathsf{M}_1)$.
 5. Toss a random coin $\eta \in \{0, 1\}$ and compute $\mathsf{Ct}^{\star} = \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct}_{\eta})$.
 6. Run $\mathcal{A}$ on input $\mathsf{Ct}_0, \mathsf{Ct}_1, \mathsf{Ct}^{\star}$ and let $\eta'$ be its output.
 7. If $\eta = \eta'$ then return 1 else return 0.
```

In $\mathsf{WUExp}_{\mathcal{A}}$ the adversary $\mathcal{A}$ selects two strings of his choice, M_0 and M_1. Both strings are encryped using different public keys (namely, Pk_0 and Pk_1) obtaining the ciphertexts Ct_0 and Ct_1, respectively. Then, one of the ciphertexts is picked at random, it is re-randomized and given to the adversary along with Ct_0 and Ct_1. We require that the adversary is not able to guess which of the two ciphertexts (i.e, tags) has been re-randomized.

Definition 3. *A tag system (*GenPub, GenKey, rEnc, rDec, $\mathsf{Randomize}$*) is* weakly untraceable *if for all probabilistic polynomial-time algorithms $\mathcal{A}$ we have that*

$$\left|\text{Prob}[\,\mathsf{WUExp}_{\mathcal{A}}(1^k) = 1\,] - \frac{1}{2}\right|$$

is negligible in k.

We remark that weak untraceability protects against adversaries that can only read tags and not write on tags. Thus it is a very weak notion and cannot be applied to our scenario of interest. In Section 4 we will give a construction of a weakly untraceable tag system which constitutes the basis for our construction of a strongly untraceable tag system.

Strong untraceability. Next we define the notion of a strongly untraceable tag system and for this we need the following experiment.

```
$\mathsf{SUExp}_{\mathcal{A}}(1^k)$
 1. $(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k)$.
 2. $(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$.
 3. Run $\mathcal{A}$ on input $\mathsf{Pub}$ and $\mathsf{Pk}$ and obtain strings $\mathsf{Ct}_0$ and $\mathsf{Ct}_1$.
 4. Set $\mathsf{Ct}_0^{\star} \leftarrow \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct}_0)$ and $\mathsf{Ct}_1^{\star} \leftarrow \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct}_1)$.
 5. If $\mathsf{Ct}_0^{\star} = \perp$ or $\mathsf{Ct}_1^{\star} = \perp$ then return 0.
```

6. Toss a random coin $\eta \in \{0,1\}$.
7. Run $\mathcal{A}$ on input $\mathsf{Ct}^\star_\eta$ and let η' be its output.
8. If $\eta = \eta'$ then return 1 else return 0.

Essentially in $\mathsf{SUExp}_\mathcal{A}$ the adversary $\mathcal{A}$ selects two strings of his choice, Ct_0 and Ct_1. Then both strings are re-randomized and, if the procedure is successful on both of them, then one is picked at random and given to the adversary. We require that the adversary is not able to guess which of the two tags has been re-randomized. Notice that if the adversary selects the two strings so that the randomization procedure fails (that is, it outputs the special failure symbol $\perp$) on exactly one of them, then traceability is unavoidable. We disallow this case by having the experiment return 0 (meaning that the adversary failed).

Observe also that the two strings Ct_0 and Ct_1 need not to be well-formed ciphertexts with respect to Pk but still the randomization procedure could be successful. However that if they both are well-formed ciphertexts then we are actually executing experiment $\mathsf{WUExp}_\mathcal{A}$. This implies that strong untraceability is stronger than weak untraceability (as one would expect).

Definition 4. *A tag system (*$\mathsf{GenPub}, \mathsf{GenKey}, \mathsf{rEnc}, \mathsf{rDec}, \mathsf{Randomize}$*) is* strongly *untraceable if for all probabilistic polynomial-time algorithms $\mathcal{A}$ we have that*

$$\left|\mathrm{Prob}[\,\mathsf{SUExp}_\mathcal{A}(1^k) = 1\,] - \frac{1}{2}\right|$$

is negligible in k.

Strong semantic security. We observe that the notion of semantic security does not make any security guarantee with respect to randomizers. In other words, randomizers are assumed to be trusted. If this is the case, then we have a very simple and direct construction of strongly untraceable tag systems. Roughly speaking, the randomizer decrypts the ciphertext and re-encrypts it using fresh randomness. If instead randomizers cannot be assumed to be trustful, then we require semantic security to hold also with respect to randomizers.

$\mathsf{SSSExp}_\mathcal{A}(1^k)$

1. $(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k)$.
2. $(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$.
3. Run $\mathcal{A}$ on input $\mathsf{Pub}, \mathsf{Pk}$ and rPub and obtain messages $\mathsf{M}_0, \mathsf{M}_1$.
4. Toss a random coin $\eta \in \{0,1\}$ and compute $\mathsf{Ct} = \mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M}_\eta)$.
5. Run $\mathcal{A}$ on input Ct and rPub and let η' be its output.
6. If $\eta = \eta'$ then return 1 else return 0.

Experiment $\mathsf{SSSExp}_\mathcal{A}$ differs from $\mathsf{SSExp}_\mathcal{A}$ in that in the former the adversary is given access to the re-randomizing information rPub and so it correctly models security against randomizers.

Definition 5. *A tag system (*$\mathsf{GenPub}, \mathsf{GenKey}, \mathsf{rEnc}, \mathsf{rDec}, \mathsf{Randomize}$*) is* strongly *semantic secure if for all probabilistic polynomial-time algorithms $\mathcal{A}$ we have that*

$$\left|\mathrm{Prob}[\,\mathsf{SSSExp}_{\mathcal{A}}(1^k) = 1\,] - \frac{1}{2}\right|$$

is negligible in k.

Finally, we have

Definition 6. *A quintuple of algorithms (*GenPub*,* GenKey*,* rEnc*,* rDec*,* Randomize*) is an* untraceable tag system *if it is strongly untraceable and strongly semantic secure.*

3 Background on Bilinear Groups

The symmetric bilinear setting. We have multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p and a non-degenerate pairing function $\mathbf{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. That is, for all $g \in \mathbb{G}$, $\mathbf{e}(g, g) \neq 1$ and $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{ab}$. We denote by g and $\mathbf{e}(g, g)$ generators of $\mathbb{G}$ and $\mathbb{G}_T$, respectively. We call a *symmetric bilinear* instance a tuple $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ and assume that there exists an efficient generation procedure $\mathcal{G}$ that, on input 1^k, outputs an instance with $|p| = \Theta(k)$.

In our constructions we make the following hardness assumptions.

Bilinear Decision Diffie-Hellman. Given a tuple $[\mathcal{I}, g^{z_1}, g^{z_2}, g^{z_3}, Z]$ for random exponents $z_1, z_2, z_3 \in \mathbb{Z}_p$ it is hard to distinguish between $Z = \mathbf{e}(g, g)^{z_1 z_2 z_3}$ and a random Z from $\mathbb{G}_T$. More specifically, for an algorithm $\mathcal{A}$ we define experiment $\mathsf{BDDHExp}_{\mathcal{A}}$ as follows.

$\mathsf{BDDHExp}^{\mathcal{A}}(1^k)$

```
Choose instance $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ running $\mathcal{G}$ with security parameter $1^k$;
Choose $z_1, z_2, z_3 \in \mathbb{Z}_p$ at random;
Choose $\eta \in \{0, 1\}$ at random;
if $\eta = 1$ then choose $z \in \mathbb{Z}_p$ at random
                  else set $z = z_1 z_2 z_3$;
Set $Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_3 = g^{z_3}$ and $Z = \mathbf{e}(g, g)^z$;
Let $\eta' = \mathcal{A}(\mathcal{I}, Z_1, Z_2, Z_3, Z)$;
if $\eta = \eta'$ then return 1 else return 0.
```

Assumption 1 (Bilinear Decision Diffie-Hellman (BDDH)). *For all probabilistic polynomial-time algorithms $\mathcal{A}$,*

$$\left|\mathrm{Prob}[\,\mathsf{BDDHExp}^{\mathcal{A}}(1^k) = 1\,] - 1/2\right|$$

is negligible in k.

Decision Linear. Given a tuple $[g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z]$ for random random exponents $z_1, z_2, z_3, z_4 \in \mathbb{Z}_p$ it is hard to distinguish between $Z = g^{z_3 + z_4}$ and a random Z from $\mathbb{G}$.

More specifically, for an algorithm $\mathcal{A}$ we define experiment $\mathsf{DLExp}_{\mathcal{A}}$ as follows.

```
$\mathsf{DLExp}^{\mathcal{A}}(1^k)$
Choose instance $\mathcal{I} = [p, \mathbb{G}, g, \mathbf{e}]$ running $\mathcal{G}$ with with security parameter $1^k$;
Choose $z_1, z_2, z_3, z_4 \in \mathbb{Z}_p$ at random;
Choose $\eta \in \{0, 1\}$ at random;
if $\eta = 1$ then choose $z \in \mathbb{Z}_p$ at random
          else set $z = z_3 + z_4$;
Set $Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_{13} = g^{z_1 z_3}, Z_{24} = g^{z_2 z_4}$, and $Z = g^z$;
Let $\eta' = \mathcal{A}(\mathcal{I}, Z_1, Z_2, Z_{13}, Z_{24}, Z)$;
if $\eta = \eta'$ then return 1 else return 0.
```

Assumption 2 (Decision Linear (DL)). *For all probabilistic polynomial-time algorithms $\mathcal{A}$,*

$$\left|\mathrm{Prob}[\,\mathsf{DLExp}^{\mathcal{A}}(1^k) = 1\,] - 1/2\right|$$

is negligible in k.

Note that Symmetric Decision Linear implies Symmetric Decision BDDH and the Symmentric Decision Linear assumption has been used in [3].

4 A First Construction

In this section we present our construction of a tag system

$$\mathsf{Tag} = (\mathsf{GenPub}, \mathsf{GenKey}, \mathsf{rEnc}, \mathsf{rDec}, \mathsf{Randomize})$$

and then we show that it is semantically secure and weakly untraceable.

4.1 The Construction

Procedure $\mathsf{GenPub}(1^k)$. We now describe the procedure GenPub used by CA to generate the public information Pub, the re-randomizing information rPub and the master secret key Msk.

1. Run $\mathcal{G}(1^k)$ to select a random bilinear instance $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ with $|p| = \Theta(k)$.
2. Pick $t_1, t_2, t_3, \omega, \in \mathbb{Z}_p$ and $g_0, g_1 \in \mathbb{G}$ at random.
3. Set
$$\Omega = \mathbf{e}(g, g)^{\omega t_1 t_2 t_3}, T_1 = g^{t_1}, T_2 = g^{t_2}, T_3 = g^{t_3}\,.$$
4. Set
$$\mathsf{Pub} = \left[\mathcal{I}, g_0, g_1, \Omega, T_1, T_2, T_3\right], \quad \mathsf{rPub} = \emptyset, \quad \text{and} \quad \mathsf{Msk} = (t_1, t_2, t_3, w).$$
5. Return $[\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}]$.

Procedure $\mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$. We now describe the procedure used by CA to generate the pair of public and secret key.

1. Pick $r \in \mathbb{Z}_p$ at random.
2. Set $\mathsf{Pk} = g_0 g_1^r$.
3. Set
$$\begin{array}{ll} D_0 = g^{rt_1t_2t_3}, & D_1 = g^{-wt_1t_3}\mathsf{Pk}^{-rt_1t_3}, \\ D_2 = g^{-wt_1t_2}\mathsf{Pk}^{-rt_1t_2}, & D_3 = g^{-wt_2t_3}\mathsf{Pk}^{-rt_2t_3} \end{array}.$$
4. Set
$$\mathsf{Sk} = [D_0, D_1, D_2, D_3].$$
5. Return $[\mathsf{Pk}, \mathsf{Sk}]$.

Procedure $\mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$. We first describe the *basic* encryption procedure $\mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ that takes as input the public parameters Pub, the public key Pk, and a cleartext $\mathsf{M} \in \mathbb{G}_T$. Then, we describe the *randomizable* encryption procedure rEnc in terms of E.

$\mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ is computed by picking $s, s_1, s_2 \in \mathbb{Z}_p$ at random and setting

$$C' = \Omega^s \cdot \mathsf{M},\ C_0 = \mathsf{Pk}^s,\ C_1 = T_2^{s_2},\ C_2 = T_3^{s-s_1-s_2},\ C_3 = T_1^{s_1}$$

$\mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ returns $[C', C_0, C_1, C_2, C_3]$.

We will use the writing $C = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M}; s, s_1, s_2)$ to denote the ciphertext computed using s, s_1 and s_2 as random choices. $\mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ simply computes

$$C = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M}) \text{ and } U = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1)$$

and returns $[C, U]$.

Procedure $\mathsf{rDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct})$. As for the encryption procedure we first describe the *basic* decryption procedure $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C)$. Let $C = [C', C_0, C_1, C_2, C_3]$ be a ciphertext. Then $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C)$ returns

$$C' \cdot \mathbf{e}(C_0, D_0) \cdot \mathbf{e}(C_1, D_1) \cdot \mathbf{e}(C_2, D_2) \cdot \mathbf{e}(C_3, D_3).$$

Simple algebra shows that if, $(\mathsf{Pk}, \mathsf{Sk})$ are a pair of public and secret keys output by $\mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$ and $C = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ then $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C) = \mathsf{M}$. Indeed, we notice that

$$\begin{aligned}
\mathbf{e}(C_0, D_0) &= \mathbf{e}(\mathsf{Pk}^s, g^{rt_1t_2t_3}) \\
&= && \mathbf{e}(g, \mathsf{Pk})^{rt_1t_2t_3s} \\
\mathbf{e}(C_1, D_1) &= \mathbf{e}(g^{t_2s_2}, g^{-wt_1t_3}\mathsf{Pk}^{-rt_1t_3}) \\
&= \mathbf{e}(g, g)^{-wt_1t_2t_3s_2} && \cdot\, \mathbf{e}(g, \mathsf{Pk})^{-rt_1t_2t_3s_2} \\
\mathbf{e}(C_2, D_2) &= \mathbf{e}(g^{t_3(s-s_1-s_2)}, g^{-wt_1t_2}\mathsf{Pk}^{-rt_1t_2}) \\
&= \mathbf{e}(g, g)^{-wt_1t_2t_3(s-s_1-s_2)} && \cdot\, \mathbf{e}(g, \mathsf{Pk})^{-rt_1t_2t_3(s-s_1-s_2)} \\
\mathbf{e}(C_3, D_3) &= \mathbf{e}(g^{t_1s_1}, g^{-wt_2t_3}\mathsf{Pk}^{-rt_2t_3}) \\
&= \mathbf{e}(g, g)^{-wt_1t_2t_3s_1} && \cdot\, \mathbf{e}(g, \mathsf{Pk})^{-rt_1t_2t_3s_1}
\end{aligned}$$

and thus

$$\mathbf{e}(C_0, D_0) \cdot \mathbf{e}(C_1, D_1) \cdot \mathbf{e}(C_2, D_2) \cdot \mathbf{e}(C_3, D_3) = \mathbf{e}(g, g)^{-wt_1 t_2 t_3 s} = \Omega^{-s}.$$

Hence,

$$C' \cdot \mathbf{e}(C_0, D_0) \cdot \mathbf{e}(C_1, D_1) \cdot \mathbf{e}(C_2, D_2) \cdot \mathbf{e}(C_3, D_3) = \Omega^{s} \cdot \mathsf{M} \cdot \Omega^{-s} = \mathsf{M}.$$

The randomizable decryption algorithm $\mathsf{rDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct})$ with $\mathsf{Ct} = [C, U]$ simply returns $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C)$.

Procedure $\mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Ct})$*.* We now describe procedure $\mathsf{Randomize}$ used to randomize a ciphertext.

A ciphertext $\mathsf{Ct} = [C, U]$ for key Pk is composed of a basic encryption C of $\mathsf{M} \in \mathbb{G}_T$ and of a basic encryption U of $1 \in \mathbb{G}_T$. Notice that $C \cdot U$ (componentwise multiplication) is a new valid basic encryption of M w.r.t. key Pk. Moreover let $U = [U', U_0, U_1, U_2, U_3]$ be a basic encryption of 1 w.r.t. key Pk. Then, for random $r, r_3, r_2 \in \mathbb{Z}_p$, $U^\star = [U'^r, U_0^r, U_1^r T_2^{r_2}, U_2^r T_3^{r_3}, U_3^r T_1^{-r_2-r_3}]$ is a randomly distributed encryption of 1 w.r.t. the same key.

Therefore, to randomize $\mathsf{Ct} = (C, U)$ we compute $(\widehat{C}, U^{\star\star})$ where $\widehat{C} = C \cdot U^\star$ and $U^{\star\star} = (U^\star)^\star$; that is, we apply the randomization of U twice and use the intermediate result $U^\star$ to randomize C. Notice that we do not need to know the public key for which C is intended.

The next lemma holds.

Lemma 1. *Assume the BDDH assumption. Then tag system* Tag *is semantically secure.*

Due to space limit, all proof are omitted and they can be found in the full version of this paper [2].

4.2 Weak Untraceability

To prove weak untraceability we show that under the Decision Linear assumption, if we apply the randomization procedure to any ciphertext $\mathsf{Ct} = [C, U]$ we obtain a tuple that is indistinguishable from a random tuple chosen from $(\mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G})^2$. We observe that it is actually enough to prove that for any basic encryption $U = [U', U_0, U_1, U_2, U_3]$ of 1 the tuple $U^{\star\star}$ is indistinguishable from a tuple chosen at random from $\mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$.

We proceed in two steps.

The first step. We prove that the following two distributions are indistinguishable under the BDDH. For any $\mathsf{M} \in \mathbb{G}_T$, define distribution $\mathsf{Dist}_0(1^k)$ as follows:

$$\begin{aligned}
\mathsf{Dist}_0(1^k) = \{ & (\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k); \\
& (\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk}); \\
& U \leftarrow \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1); \\
& C \leftarrow \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M}); \\
& [\widehat{C}, U^{\star\star}] \leftarrow \mathsf{Randomize}(\mathsf{Pub}, \mathsf{rPub}, [C, U]) : (\mathsf{Pub}, U, U^{\star\star}) \}
\end{aligned}$$

while, distribution $\mathsf{Dist}_1(1^k)$ is defined as follows:

$$\begin{aligned}\mathsf{Dist}_1(1^k) = \{&(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k);\\ &(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk});\\ &U \leftarrow \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1);\\ &r, s', r_2, r_3 \leftarrow \mathbb{Z}_p;\\ &U^\star = [\Omega^{rs'}, U_0^r, U_1^r T_2^{r_2}, U_2^r T_3^{r_3}, U_3^r T_1^{-r_2-r_3}] : (\mathsf{Pub}, U, U^\star)\}\end{aligned}$$

In the definition of Dist_1 we have denoted by Ω, T_1, T_2, T_3 the components of Pub and by U', U_0, U_1, U_2, U_3 the components of U. Notice that if we write U as $U = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1; s, s_1, s_2)$ then we have

$$U^\star = [\Omega^{rs'}, \mathsf{Pk}^{rs}, T_2^{rs_2+r_2}, T_3^{r(s-s_1-s_2)+r_3}, T_1^{rs_1-r_2-r_3}].$$

That is, $U^\star$ is a ciphertext for a random element of $\mathbb{G}_T$ for public key Pk (specifically, $U^\star$ is an encryption of $\Omega^{rs'-rs}$). Indistinguishability of Dist_0 and Dist_1 can be argued by a reasoning similar to the one employed to prove semantic security (see [2] for a complete proof).

The second step. We can prove that, under the Decision Linear assumption, distributions

$$\begin{aligned}\mathsf{Dist}_2(1^k) = \{&(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k);\\ &(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk});\\ &U \leftarrow \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1);\\ &s, s', r_2, r_3 \leftarrow \mathbb{Z}_p;\\ &U^\star = [\Omega^{s'}, \mathsf{Pk}^{s}, T_2^{r_2}, T_3^{r_3}, T_1^{s-r_2-r_3}] : (\mathsf{Pub}, U, U^\star)\}\end{aligned}$$

and

$$\begin{aligned}\mathsf{Dist}_3(1^k) = \{&(\mathsf{Pub}, \mathsf{rPub}, \mathsf{Msk}) \leftarrow \mathsf{GenPub}(1^k);\\ &(\mathsf{Pk}, \mathsf{Sk}) \leftarrow \mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk});\\ &U \leftarrow \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1);\\ &s, s', r_1, r_2, r_3 \leftarrow \mathbb{Z}_p;\\ &U^\star = [\Omega^{s'}, \mathsf{Pk}^{s}, T_2^{r_2}, T_3^{r_3}, T_1^{r_1}] : (\mathsf{Pub}, U, U^\star)\}\end{aligned}$$

are indistinguishable. Notice that Dist_2 is just a re-writing of Dist_1 and that Dist_3 is the random distribution on $\mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$. The second step completes the proof that $U^{\star\star}$ is indistinguishable from a random quintuple. We have the following lemma.

Lemma 2. *Assume the Decision Linear assumption. Then tag system* Tag *is weakly untraceable.*

Why strong untraceability is not guaranteed. The scheme described in this section is only weakly untraceable. Let us see where our proof breaks for strong untraceability. The first step of the proof essentially says that distribution $(\Omega^s, \mathsf{Pk}^s)$ with random $s \in \mathbb{Z}_p$ is indistinguishable from distribution $(\Omega^{s'}, \mathsf{Pk}^s)$ with random $s, s' \in \mathbb{Z}_p$. The analogous statement for the case of strong untraceability would have been, for any $A \in \mathbb{G}_T$ and $B \in \mathbb{G}$ ((A, B) is one of the strings given in output by the adversary $\mathcal{A}$ at step 3 of $\mathsf{SUExp}_{\mathcal{A}}$), the distribution (A^s, B^s) with random $s \in \mathbb{Z}_p$ (this is the distribution of the output of the randomizer on input (A, B)) is indistiguishable from the distribution (A^s, B^r) with random $r, s \in \mathbb{Z}_p$ (this is the random distribution on $\mathbb{G}_T \times \mathbb{G}$).

It is easy to see that if A and B are adversarially chosen the assumption is false. In fact, the adversary may choose $A = \mathbf{e}(a, a)$ for random $a \in \mathbb{G}$ and $B = a^b$ for random $b \in \mathbb{Z}_p$. Then, for any $s \in \mathbb{Z}_p$ and for $(C, D) = (A^s, B^s)$, we have $\mathbf{e}(a, D) = C^b$. On the other hand, for random $r, s \in \mathbb{Z}_p$ if $(C, D) = (A^s, B^r)$ then $\mathbf{e}(a, D) = C^b$ with negligible probability.

5 Strong Untraceability

In this section we present a transformation that takes the weakly intraceable tag system $\mathsf{Tag} = (\mathsf{GenPub}, \mathsf{GenKey}, \mathsf{rEnc}, \mathsf{rDec}, \mathsf{Randomize})$ of the previous section and tranforms it into a strongly untraceable tag system

$$\mathsf{STag} = (\mathsf{SGenPub}, \mathsf{SGenKey}, \mathsf{SrEnc}, \mathsf{SrDec}, \mathsf{SRandomize}).$$

5.1 The Transformation

Our transformation employs a regular semantically-secure encryption scheme $\mathcal{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$.

Procedure $\mathsf{SGenPub}(1^k)$. Execute procedure $\mathsf{GenPub}(1^k)$ and obtain $[\mathsf{Pub}, \emptyset, \mathsf{Msk}]$. Then, execute the key-generation procedure KG of the secure encryption scheme $\mathcal{E}$ and obtain $(\mathsf{rpk}, \mathsf{rsk})$. The output of the procedure is the triple $[\mathsf{SPub}, \mathsf{SrPub}, \mathsf{SMsk}]$ where

$$\mathsf{SPub} = (\mathsf{Pub}, \mathsf{rpk}), \quad \mathsf{SrPub} = \mathsf{rsk}, \quad \text{and} \quad \mathsf{SMsk} = \mathsf{Msk}.$$

Procedure $\mathsf{SGenKey}(\mathsf{SPub}, \mathsf{SMsk})$. The key generation procedure takes as input the public information $\mathsf{SPub} = (\mathsf{Pub}, \mathsf{rpk})$ and the master secret key $\mathsf{SMsk} = \mathsf{Msk}$, invokes $\mathsf{GenKey}(\mathsf{Pub}, \mathsf{Msk})$ to obtain $[\mathsf{Pk}, \mathsf{Sk}]$, and returns $[\mathsf{Pk}, \mathsf{Sk}]$.

Procedure $\mathsf{SrEnc}(\mathsf{SPub}, \mathsf{Pk}, \mathsf{M})$. The encryption procedure SrEnc takes as input the public information $\mathsf{SPub} = (\mathsf{Pub}, \mathsf{rpk})$, the public key Pk, and a cleartext M, invokes[1] $\mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ to obtain C, and returns the ciphertext $\mathsf{Ct} = [C, \mathsf{Enc}(\mathsf{Pk}, \mathsf{rpk})]$.

Procedure $\mathsf{SrDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct})$. The decryption procedure SrDec takes as input the public information $\mathsf{SPub} = (\mathsf{Pub}, \mathsf{rpk})$, the private key Sk, and the ciphertext $\mathsf{Ct} = [C_0, C_1]$ and returns[2] $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C_0)$.

[1] Recall that $\mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$ is the *basic* encryption procedure used in $\mathsf{rEnc}(\mathsf{Pub}, \mathsf{Pk}, \mathsf{M})$.

[2] Recall that $\mathsf{D}(\mathsf{Pub}, \mathsf{Sk}, C_0)$ is the *basic* decryption procedure used in $\mathsf{rDec}(\mathsf{Pub}, \mathsf{Sk}, \mathsf{Ct})$.

Procedure $\mathsf{SRandomize}(\mathsf{SPub}, \mathsf{SrPub}, \mathsf{Ct})$. The randomization procedure $\mathsf{SRandomize}$ takes as input the public information $\mathsf{SPub} = (\mathsf{Pub}, \mathsf{rpk})$, the randomizing information $\mathsf{SrPub} = \mathsf{rsk}$, and the ciphertext $\mathsf{Ct} = [C_0, C_1]$ and proceeds as follows.

1. Let $\mathsf{Pk} = \mathsf{Dec}(C_1, \mathsf{rsk})$. If decryption fails then return $\perp$ and halt.
2. If $C_0 \notin \mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$ then return $\perp$ and halt.
3. Compute $U = \mathsf{E}(\mathsf{Pub}, \mathsf{Pk}, 1)$.
4. Set $\widehat{C}_0$ equal to the component-wise product of C_0 and U.
5. Set $\widehat{C}_1 = \mathsf{Enc}(\mathsf{Pk}, \mathsf{rpk})$.
6. Return $(\widehat{C}_0, \widehat{C}_1)$.

We next briefly argue the security properties of the tag system STag. Strong Semantic security follows directly from the proof of semantic security of the tag system Tag (see Lemma 1 of [2]).

Let $\mathsf{Ct} = [C_0, C_1]$ be an adversarially chosen pair. We assume that $C_0 \in \mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$ and that C_1 encrypts public key Pk. If this is not the case then the $\mathsf{SRandomize}$ fails and returns $\perp$. Notice that if $\mathsf{SRandomize}$ does not return $\perp$ then C_0 is a valid encryption of a message M with respect to public information Pub and some public key Pk' (notice that we do not necessarily have that $\mathsf{Pk} = \mathsf{Pk}'$). Let $\widehat{\mathsf{Ct}} = [\widehat{C}_0, \widehat{C}_1]$ be the output of $\mathsf{SRandomize}$. Observe that by the semantic security of Enc, $\widehat{C}_1$ is indistinguishable from an encryption of a random string (of the same length as Pk). In addition, $\widehat{C}_0$ is the encryption of message M' with respect to public key Pk'. We distinguish two case. If $\mathsf{Pk} = \mathsf{Pk}'$ then $\mathsf{M}' = \mathsf{M}$ and, by the weak untraceability of tag system Tag (see Lemma 2 in [2]), $\widehat{C}_0$ is indistinguishable from a random element of $\mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$.

If $\mathsf{Pk} \neq \mathsf{Pk}'$ then $\widehat{C}_0$ is the encryption of a random element M' of $\mathbb{G}_T$ which is indistinguishable from a random element of $\mathbb{G}_T \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G}$. This follows from arguments similar to the ones used to prove the semantic security of Tag (see Lemma 1 in [2]). We thus have the following theorem.

Theorem 3. *Assume the Decision Linear assumption. The tag system* STag *is an untraceable tag system.*

6 Performances Analysis

In this section, we present the results of some experiments that we ran to evaluate the real applicability and the lightness of our schemes for untraceable tags. We also compare them with the scheme presented in [1]. For our experiments, we set up the following small test-bed:

- PC: Intel Core 2 Quad Q6600 2.40 GHz, 3 GB RAM.
- OS: Ubuntu 9.04 - kernel 2.6.28-11-generic - 64 bit.
- PBC Library ver. 0.4.18 [11].
- dcrypt Library ver. 0.3 [4].

Table 1. Execution times in milliseconds of the schemes' procedures

	`Weak`	`Strong`	`InsEnc`
Public Information Generation	53.3819	85.0649	103.6559
Key Generation	54.2332	81.1983	24.9269
Encryption	67.1600	49.7901	24.9460
Decryption	46.8101	47.0070	93.4520
Randomization	147.7423	53.9974	116.4845

Table 2. Size in bytes of the encryption

	`Weak`	`Strong`	`InsEnc`
Bytes written on tag	1520	1281	364

In the following tables we summarize the results of our experiments. The second column (e.g., `Weak`) corresponds to the scheme presented in Section 4.1; the third column (e.g., `Strong`) presents the results for the scheme satisfying the strong untraceability property (the scheme is described in Section 5.1); while, the last column (e.g., `InsEnc`) describes the results attained by the Insubvertible Encryption scheme proposed in [1].

For the tests, we set the security parameter to $k = 1206$. Tests were repeated 5000 times. We took the time needed to execute each procedure of an untraceable tag system. In Table 1 we report the average time (expressed in milliseconds) taken by the tests we ran. Considering the randomization procedure, in spite of relying on weaker assumptions, our strong scheme has a better performance, in terms of computational requirements, than the scheme presented in [1]. Our randomization procedure (as well as the decryption one) runs twice faster as the one of [1]. This is very important, as the randomization procedure is invoked quite often (e.g., each time a tag is in proximity of a randomizer); while, all other procedures are invoked just once. Moreover, the randomization procedure is run by special devices (i.e., randomizers) which have low computing power; while, the other procedures are executed by more powerful devices.

As one can see from Table 2, both our schemes generate an encrypted message (to be written on the tag) of size greater than the one generated by the scheme in [1]. This is not a big concern, as our encrypted messages easily fit in the user memory of currently produced passive RFID tags. For instance, Maxell provides RFID tags whose memory capacity ranges from 128 bytes up to 4K bytes [8]. Moreover, there exists passive RFID having user memory of 32K bytes [13].

7 Extensions and an Open Problem

Our construction of STag is a special case of a general construction that starts from a *randomizable* anonymous identity-based encryption scheme that enjoys a weak form of security (specifically, security against randomly chosen identities) and turns into an untraceable tag system. Unfortunately, no randomizable anonymous identity-based encryption was known prior to our work, and thus we had to construct our own.

The strong untraceability property defined in this paper does not give any guarantee against randomizers as in experiment SUExp adversary $\mathcal{A}$ has not access to rPub. It would be nice to give a construction which guarantees untraceability against randomizers and whose security is based on mild assumptions.

Nonetheless, as it is not difficult to see, if we use tag system STag, randomizers cannot distinguish between tags carrying encryptions computed with respect to the same public key. This is a very important property since in many applications the public key corresponds to the manufacturer of the object to which the tag is attached. An adversary thus does not need to look at the tag to distinguish objects from different manufacturers and the applicability of tag system STag is not limited.

References

1. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable rfid tags via insubvertible encryption. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM Conference on Computer and Communications Security, pp. 92–101. ACM, New York (2005)
2. Blundo, C., De Caro, A., Persiano, G.: Untraceable tags based on mild assumptions. Cryptology ePrint Archive, Report 2009/380 (2009), http://eprint.iacr.org/
3. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
4. The dcrypt Library. Ver. 0.3, http://www.scs.cs.nyu.edu/css/lab/dcrypt_fns.html
5. Garfinkel, S., Rosenberg, B.: RFID: Applications, Security, and Privacy. Addison-Wesley Professional, Reading (2005)
6. Golle, P., Jakobsson, M., Juels, A., Syverson, P.F.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
7. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems (Extended abstract). In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, p. 184. Springer, Heidelberg (2000)
8. Maxell. Coil-on-Chip RFID (2009), http://www.maxei.co.jp/products/coc/eng-smal_chip.html
9. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)
10. Near field communication forum, http://www.nfc-forum.org/
11. PBC: The Pairing-Based Cryptography Library. Ver. 0.4.18, http://crypto.stanford.edu/pbc/
12. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
13. TegoTag (2009), http://www.tegoinc.com/

Security Threat Mitigation Trends in Low-Cost RFID Systems

Joaquin Garcia-Alfaro[1,2], Michel Barbeau[1], and Evangelos Kranakis[1]

[1] School of Computer Science,
Carleton University, K1S 5B6, Ottawa, Ontario, Canada
{barbeau,kranakis}@scs.carleton.ca
[2] Open University of Catalonia,
Rambla de Poble Nou, 156, 08018, Barcelona, Spain
joaquin.garcia-alfaro@acm.org

Abstract. The design and implementation of security threat mitigation mechanisms in RFID systems, specially in low-cost RFID tags, are gaining great attention in both industry and academia. One main focus of research interests is the authentication and privacy techniques to prevent attacks targeting the insecure wireless channel of these systems. Cryptography is a key tool to address these threats. Nevertheless, strong hardware constraints, such as production costs, power consumption, time of response, and regulations compliance, makes the use of traditional cryptography in these systems a very challenging problem. The use of low-overhead procedures becomes the main approach to solve these challenging problems where traditional cryptography cannot fit. Recent results and trends, with an emphasis on lightweight techniques for addressing critical threats against low-cost RFID systems, are surveyed.

Keywords: Radio Frequency Identification (RFID), Electronic Product Code (EPC), Wireless Security, IT Security, Security Threats, Privacy Threats.

1 Introduction

Radio Frequency Identification (RFID) is a wireless communication technology, based on analog and digital components, used to identify and track goods and people. Even though it has been used for more than seventy years (e.g., RFID was used in World War II for identifying enemy aircrafts), it is only now that this technology is re-emerging as an important communication paradigm that claims to revolutionize inventory and automation processes [61]. Examples are the use of RFID for supply chain inventory, health care management, animal identification, and anti-counterfeiting. However, while this technology is gaining importance with industrial suppliers, security and privacy concerns are raising, especially, among RFID consumers and end users.

An example is the introduction of low-cost RFID technology in the supply chain of the retail industry by means of the Electronic Product Code (EPC) concept. The EPC is a unique code associated to a passive RFID tag that is placed on shipment pallets. The ability to identify and track these pallets, and their associated products, raise security and privacy concerns. These concerns become critical as retailers and manufacturers

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 193–207, 2010.

contemplate moving from pallet tagging to individual item tagging [61]. The possibility of rogue monitoring of people carrying these items is stimulating security and privacy research in both industry and academia. The insertion of cryptographic mechanisms in low-cost RFID tags is a promising solution to address the aforementioned concerns. However, current state-of-the-art solutions in cryptography must face significant challenges before being deployed in RFID technologies.

According to a research presented by Sarma in [62], the maximum cost of passive EPC tags should not exceed five cents to enable successful deployment on a world wide scale. This research also states that of these five cents, only one or two cents should be used for the manufacturing of the Integrated Circuit (IC). It is assumed that the available layout area for the implementation of the IC is in the range of 0.25mm^2 which, considering current CMOS technology, translate to a theoretical number of logic gates from two to four thousand. Not all the barriers investigated in [62] have been removed. The low-cost RFID technology of today is more expensive than what it was anticipated — around ten cents in large quantities. The inclusion of additional RFID features, especially for authentication and privacy purposes, may increase the total end-cost of tags up to fifteen cents or more per unit. Although Moore's Law predicts that digital devices fabricated on ICs will continue decreasing in price, cost of analogue devices (i.e., RF front-end of tags) will remain a constraint [12]. The inclusion of new elements must therefore be well planned.

Since the power used by low-cost RFID (passive) tags is derived from the signal received from readers, power restrictions also apply. The power consumption of a tag varies according to the nature of the operation being performed (e.g., responding to a query or writing data into the memory) and other parameters like the transmission rate, response time, and memory technology. Most of the operations performed in EPC tags require about five to ten microamps — although some special operations, such as writing operations, may require higher power. The power consumption of new security primitives must be within this range in order to allow low-cost tag production.

New security primitives must also work at the data rate of EPC applications. Current EPC applications demand an average reading speed of about two hundred tags per second. That leads to a data transmission rate requirement from tag to reader, of about 640 kbps; and a transmission rate from reader to tag of about 120 kbps. Delays associated to new security mechanisms (e.g., time to perform encryption or random number generation) may also affect the global performance. Delays must hence be taken into account and minimized. We can find in the literature several solutions that provide authentication and privacy mechanisms while meeting these challenging constraints. Most of the solutions can be classified in the following three categories: (1) lightweight cryptography based on the use of one-way hash-like primitives implemented in tags; (2) low-overhead and ultra-lightweight cryptography relying on the single use of on-tag pseudorandomness and simple arithmetic operations; and (3) alternative solutions avoiding the execution of cryptographic processes within tags. We survey, in the sequel, recent contributions and trends according to these three categories. Our work aims at increasing the awareness of available security threat mitigation methods among RFID researchers and developers.

Paper organization. Section 2 surveys one-way hash-like solutions. Section 3 surveys proposals based on the single use of on-tag pseudorandomness and simple arithmetic operations. Section 4 surveys alternative approaches not requiring the necessity of on-tag cryptographic processes. Section 5 concludes the paper.

2 Lightweight Cryptographic Approaches

MAC (Message Authentication Code) based security protocols are among the first solutions discussed in the literature for securing low-cost RFID applications. In [65], for example, Takaragi et al. present a simple MAC-based approach that uses a static unrewritable 128-bit identifier stored, at manufacturing time, in every tag. The identifier is generated by the manufacturer using a unique secret key for each tag and a keyed hash function that accepts as input the secret key and a specific message. The secret key, hash function, and specific messages are communicated by the manufacturer to the client. Then, this information is shared among the clients' readers, to verify the integrity and authenticity of the exchanged messages. Therefore, this mechanism increases the technical difficulties of performing attacks against the integrity and authenticity of the messages. The main drawback is the use of static identifiers embedded in the tags at manufacturing time. Therefore, brute force attacks can break the secrets shared between readers and tags.

An enhanced solution relies on the use of hash-lock schemes for implementing access controls. In [69], Weis et al. propose a way to prevent unauthorized readers from reading tag contents. A secret is sent by authorized readers to tags using a trusted environment. Tags, equipped with an internal hash function, perform a hash on this secret and store it within their internal memory. Then, tags enter into a locked state in which they answer to any possible query with the computed hash. Weis et al. also describe proper ways of unlocking tags, if such an action is needed by authorized readers (i.e., to temporarily release private data). Regarding privacy threats, Ohkubo et al. propose in [49] the use of hash chains for the implementation of on-tag security mechanisms with evolving RFID identities. Avoine and Oechslin discuss in [2] some limitations of the approach. They propose an enhanced hash-based RFID protocol to address both authentication and privacy by using timestamps. Similarly, Henrici and Müller discuss in [25] some weaknesses in the hash-lock scheme presented in [69] and propose a new hash-based scheme intended to enhance privacy and authentication. Several other improvements and hash-based protocols, most of them inspired on lightweight cryptography research for devices with higher hardware capabilities such as smart cards, can be found in [48,16,43,53].

2.1 Hardware Challenges and Limitations

Let us note that the aforementioned approaches require the implementation of one-way hash primitives within low-cost RFID tags. The requirement of reliable hash primitives implemented at the tag level is the main challenge associated with these proposals. Gate requirements of implementations based on standard one-way hash functions, such as MD4, MD5, and SHA-128/SHA-256, exceed the constraints pointed out in Section 1.

The implementation of these functions require from seven thousand to over ten thousand logic gates; and from six hundred to over one thousand two hundred clock cycles [53]. The complexity of standard one-way hash functions is therefore an impediment for their deployment on low-cost RFID tags.

The use of standard encryption engines for the construction of hash operations has been discussed in the literature. For example, the use of Elliptic Curve Cryptosystems (ECC) [47] for the implementation of one-way hash primitives on RFID tags has been studied in [72]. Its use of small key sizes is seen as very promising for providing an adequate level of computational security at a relatively low cost [12]. An ECC implementation for low-cost RFID tags can be found in [4]. In [21], Feldhofer et al. present a 128-bit implementation of the Advanced Encryption Standard (AES) [14] on an IC of about three thousand five hundred gates with a power consumption of less than nine microamps at a frequency of 100 kHz. Although this implementation is considerably simpler than previous implementations of the AES algorithm, its requirements are still too high for low-cost RFID tags.

Alternative hash functions based on non-standard low-cost encryption engines is a third candidate. In [29], Israsena presents a hardware implementation of the Tiny Encryption Algorithm (TEA) [70] on an IC of about three thousand gates and with a consumption of about seven microamps. It fits the timing requirements of basic EPC setups where hundred of tags must simultaneously be accessed by the same reader. The implementation relies on very simple arithmetic and bitwise operators. The authors of TEA [70] claim that, despite its simplicity and ease of implementation, the complexity of the algorithm is equivalent to the one of DES (Data Encryption Standard) [47]. Variants of the TEA algorithm are, however, necessary for implementing hash functions. Mace et al. discuss in [46] some of the vulnerabilities of TEA, such as linear and differential cryptanalysis attacks, and present SEA (Scalable Encryption Algorithm). The strength of this proposal, due to its novelty, is not clear [12].

Other low-cost alternatives are the single use of Linear and Non Linear Feedback Shift Registers (LFSR & NLFSR). However, the simple use of LFSR & NLFSR as underlying mechanisms for the implementation of low-cost one-way hash functions — without further measures that add cost of extra hardware — lead to insecure implementations. For example, the use of the Cellular Automata (CA) model [71] for the implementation of one-way functions — typically built upon LFSR & NLFSR — has been proved to lead to insecure implementations [5,12].

2.2 Physical One-Way Functions

The design of Physically Unclonable Functions (PUFs) and Physical Obfuscated Keys (POKs) is promising for the implementation of hash-like protocols on low-cost EPC tags. Half way between traditional cryptography and physical protection defenses, the ideas behind PUFs and POKs originated in [51] with the conception of optical mechanisms for the construction of Physical One-Way Functions (POWFs). Its use to securely store unique secret keys, in the form of fabrication variations, was proposed as a silicon prototype in [23]. The ideas were later improved in [45]. A coating PUF proposed in [64] claims an implementation that requires less than one thousand gates. The designs exploit the random variations in delays of wires and logic gates of an IC. For example, the silicon PUF presented in [23] receives input data, as a challenge, and launches a race condition within the IC: two transitions signals start propagating along different paths and are compared to determine which one comes first. To decide which signal comes first, a special controller produces a binary value.

The implementation of these proposals seems to have clear advantages at a cost of less than one thousand logic gates [64]. This technology provides a cost effective and reliable solution that successfully meet the constraints and requirements mentioned in Section 1. However, it also has several drawbacks. The difficulty of successfully modeling the circuits and their reliability is one of the obstacles that this technology must to face. The effects of environmental conditions and effects of the power supply voltage have also raised some concerns [12]. Some alternative proposals try to solve the drawbacks. Holcomb et al. propose in [26] an approach based on the CMOS SRAM memory of an electrical to generate physical fingerprints. The key idea is the usage of SRAM startup values as seeds of pseudorandomness. The authors claim that the use of 256 bytes of SRAM can yield 100 bits of true randomness each time the memory is powered up. While sound in theory, this technique is limited by memory space of current low-cost tags.

Challenge-response protocols are commonly used to implement security mechanisms in low-cost RFID tags using PUFs. An initial approach presented in [58], and based on PUFs proposed in [23], consists of a challenge-response scheme that probabilistically ensures unique identification of RFID tags. The back-end system of this approach must learn challenge-response pairs for each PUF/tag. It then uses these challenges (hundreds of them) at a time, to identify and authenticate tags. Unique identification of tags is probabilistic. The exposition of tag identifiers to eavesdroppers, and lack of state and randomness in tag responses, make the approach vulnerable to tracking and location threats. Moreover, the great number of challenges that are necessary between readers and tags for the completion of the identification process increases tag response delay and power consumption.

An alternative protocol is presented in [67]. Tuyls and Batina discuss an off-line PUF-based mechanism for verifying the authenticity of tags using the PUF technology presented in [64]. Similarly to the traditional approaches presented in [31,36], where readers and tags define ad hoc secrets, the PUF-based approach uses instead the physical structure embedded within tags to generate unique keys. A key extraction algorithm from noisy (binary) data is presented in [67]. The usage of PUF-based keys simplifies the process of verifying tag authenticity. The combination of unique keys generated onboard together with public key cryptography techniques (e.g., use of signatures) avoid leaking the static single identifier and hence increase the technical difficulties for an attacker to carry on location tracking threats. The main drawback of the approach is the need of large storage space and reliable searching processes on back-end servers in order to link readers with PUF/tag identifiers. The use of public key and digital signatures, based on Elliptic Curve Cryptography (ECC), is another important constraint of the approach.

Bolotnyy and Robins propose in [6] a complete set of adapted MAC protocols based on PUFs aiming to simplify the challenge-response communication scheme of previous proposals, and the requirement of traditional cryptographic primitives. Each tag generates multiple identifiers based on embedded PUFs. The approach only addresses static identification and is vulnerable to location tracking attacks. The approach does not solve the necessity of huge lists of challenge-response pairs for each PUF/tag which must be stored on back-servers connected to the readers. Indeed, once a given pair is sent, it must not be used anymore. Otherwise, the protocol cannot guarantee that an adversary eavesdropping data will not gain advantage by performing a replay attack.

3 Low-Overhead and Ultra-Lightweight Solutions

The use of on-tag Pseudo Random Numbers Generators (PRNGs) to enhance the security and privacy of RFID systems is another candidate. In fact, most of the approaches, if not all, presented in Section 2 require the use of PRNGs to guarantee correctness. For example, the enhanced hash-lock scheme presented by Weis et al. in [69] relies on the use of on-tag PRNGs and efficient pseudorandomness for mitigating privacy threats like location tracking. Another example is the need of combining PRNGs and hash chains to enable the proposal of Ohkubo et al. presented in [49]. More recently, a protocol presented in [66], called YA-TRAP, reduces the need of hash-based protocols by combining pre-computed hash-tables for tag verification processes with timestamps and generation of pseudorandom numbers. Similar requirements apply on all the other protocols surveyed in Section 2 — in order to address location tracking problems. From a hardware point-of-view, the insertion of robust one-way hash functions and PRNGs in the constrained environment of low-cost RFID tags makes the implementation of those proposals very challenging and, unrealistic for real world applications.

The use of pseudorandomness for increasing low-cost RFID security is often questioned because robust designs are complex to implement on low-cost RFID devices. The complexity of the implementation of robust PRNGs is equivalent to the complexity of the implementation of robust one-way hash-functions and/or equivalent encryption engines [47]. However, since the ratification of the EPCglobal standard EPC Class-1 Generation-2 (Gen2 for short) [20] and ISO standards ISO/IEC 18000-6C [28] for the usage of on-tag PRNGs on low-cost RFID devices, the number of single PRNG-based solutions has increased in the industry and academia research. The existence of PRNG hardware already deployed on most of the low-cost RFID tags justifies the convenience of this second category of security threat mitigation mechanisms.

Juels and Weis present in [36] an unidirectional authentication protocol based on the secure human identification protocol series proposed by Hopper and Blum [27]. The new protocol, called by the authors HB+, aims at preventing active attacks against the authenticity of low-cost RFID systems. The resistance of HB+ against active adversaries is proved by the authors using an statistical conjecture [13] to bound the difficulty of learning a secret (e.g., ID of the tag) given a sequence of randomly chosen vectors with embedded noisy information. The authors claim that the protocol can be implemented on low-cost tags since it only requires PRNG primitives in tags and implementation of very simple operations, such as bitwise-and and xor. Some security issues of the HB+ protocol were reported in [39,57]. They propose enhancements to address active attacks. However, neither the original HB+ protocol nor its sequels consider authentication of the readers and location tracking attacks. Regarding these issues, we can find in [38] a new low-overhead protocol by Karthikeyan and Nesterenko for mutual authentication of tags and readers. The requirements of this protocol are modular algebra operations, such as multiplication of matrices, and on-tag PRNG primitives. Based on similar requirements, such as on-tag PRNG and matrix algebra operations, Dolev et al. present in [17] two low-overhead proactive unidirectional protocols, called PISP (Proactive Informational Secure Protocol) and PCSP (Proactive Computationally Secure Protocol), with evolving on-tag secrets that expands indefinitely over time. Both PISP and PCSP are compared and contrasted in a joint publication appeared in [19]. The security of these protocols relies on the difficulty of recovering the operands used on both sides (tags and readers) to synchronize shared secrets. Memory space on current low-cost tags is another limitation to the security of these approaches. An enhanced version of the PCSP protocol, presented in [18], aims at preventing active attacks against the protocol while keeping similar requirements, i.e., on-tag PRNG primitives and matrix operations.

Burmester, Le, and de Medeiros proposed in [7] a new low-overhead protocol, called O-TRAP (Optimistic Trivial RFID Authentication Protocol). Like other protocols surveyed in this section, O-TRAP relies on the use of PRNG primitives in tags and some other simple bitwise operations. O-TRAP is specially designed to prevent privacy attacks while guaranteeing anonymous authentication. The protocol behaves in a manner similar to the hash-lock approach introduced in Section 2. Common secret, shared between readers and tags, are proposed in their scheme to update pseudonyms stored within tags. Like in the hash-lock approach introduced by Weis et al. in [69], readers must access back-end databases to map pseudonyms to true identities. The security of the protocol is proved using the universal composability (UC) model [8]. It is shown that the O-TRAP protocol meets the UC definition of anonymous authentication and anonymous key exchange. However, the O-TRAP protocol fails to satisfy the stronger privacy definitions, such the one stated by Juels and Weis in [37] establishing that privacy countermeasures must guarantee both anonymity and untraceability. Juels and Weis point out the possibility of attacking the O-TRAP protocol by de-synchronizing tags. This allow active attacker to uniquely identify them and carry on location tracking attacks. An attack against the untraceability of the O-TRAP protocol is presented in [50].

Similar attacks exploit existing vulnerabilities in the state-of-the-art of the ultra-lightweight series of authentication protocols. Ultra-lightweight authentication protocols, such as [54,55,56,11], try to eliminate the necessity of hash and PRNG primitives, and involve only simple bitwise and modular arithmetic on-tag operations. The computation of costly operations, such as the generation of pseudorandom numbers, is done at the reader side. Although this fact benefits the implementation of such countermeasures on the constrained environment of low-cost RFID tags, none of these proposals seems to be resistant to either active or passive attacks. The set of authentication techniques presented by Peris-Lopez et al. in [54,55,56] were reported to be vulnerable by Li and Wang, and Li and Deng to, respectively, the de-synchronization attacks and full-disclosure attacks. Improvements of these techniques, presented by Chien in a new protocol called SASI [11] have recently been reported as vulnerable by Cao, Bertino, and Lei in [9]. These recent cases show how challenging it is to design adequate procedures given the low-cost requirement of the RFID paradigm.

4 Avoidance of On-Tag Cryptographic Processes

Several results, such as [32,40,30,31], are not relying on the execution of cryptographic algorithms in tags. One of the earliest proposals is the re-encryption scheme of Juels and Pappu presented in [32]. It provides privacy and security for banknotes embedding RFID tags. The approach uses public key cryptography and digital signatures. The operations are, although, performed outside the tags. The scheme consists of a public-key cryptosystem and two authorities: a central bank and a law enforcement agency. Both authorities hold an independent pair of public and private keys associated to each banknote. The central bank authority assigns a unique serial number to each banknote. To do so, the bank uses its private key to sign the unique serial number. The signature and the serial number of the banknote are printed on the banknote as optical data. Then, by using the public key of the law enforcement agency, the bank encrypts the digital signature, unique serial number, and a random number. The resulting ciphertext is stored into a memory cell of the RFID tag. This memory cell is keyed-protected. The tag only grants write access to this memory cell if it receives an access key derived from the optical data. The random number used to create the ciphetext is also stored into a separated memory cell of the tag. This second memory cell is also keyed-protected. The tag only grants read or write access to this memory cell if it receives an access key derived from the optical data.

By using this previous approach, banknote bearers must verify first the digital signature, printed in the banknote as optical data, using the public key of the central bank. Second, they must also verify the validity of the ciphertext stored in the RFID tag. To do so, the bearer encrypts the digital signature, serial number, and random number stored in the memory of the tag, using the public key of the law enforcement agency and the optical data. If one of these two verification processes fails, the authorities must be warned. To avoid using the same ciphertext on every interaction, the authors propose the use of a re-encryption process that can be performed by banknote bearers without the necessity of accessing the private keys of the law enforcement authority. Based on the algebraic properties of the El Gamal cryptosystem [47], the initial ciphertext is transformed into a new unlinkable ciphertext, using the public key of the law enforcement authority [32]. This re-encryption process is performed outside the tags. Although the whole process is too complex for use in low-cost RFID scenarios, it is one of the first solutions that appeared in the literature for deploying cryptographic protocols in RFID applications without the need to embed cryptographic primitives in tags.

The work presented by Kinosita et al. in [40] consists of an anonymous ID scheme, in which a tag contains only a pseudonym that is periodically rewritten. Similarly, the approach of Juels in the work *Minimalist Cryptography for Low-Cost RFID Tags* [30] suggests a very light-weight protocol for mutual authentication between tags and readers based on one-time authenticators. Both solutions rely on the use of pseudonyms and keys stored within tags and back-end servers. Pseudonyms are used instead of real identifiers (e.g., instead of the EPC codes in supply chain RFID applications). Each tag contains a small collection of pseudonyms, according to the available memory. A throttling process, is used to rotate these pseudonyms. Each time a tag is interrogated by a reader, a different pseudonym selected at random is returned. Authorized readers have access to the complete list of pseudonyms of each tag and can correlate the identity of the responses they receive. Without the knowledge of this list, unauthorized readers are unable to infer any information about the numerous occurrences of the same tag. The process also forces tags to slow their transmissions when queries come too quickly, as a defense to brute-force attacks. The memory space in current low-cost tags is the main limitation of this approach. Although enhancements can be used to update the list of pseudonyms, communication costs and integrity threats still remain as main drawbacks. A similar, though lighter-weight, protocol for mutual authentication between readers and tags is presented by Juels in [31]. This time, the Personal Identification Number (PIN), associated to the kill command of EPC Gen2 tags [20], is used to implement the protocol. The main idea is that even if the EPC data of a tag is skimmed, the PIN remains secret. This way, cloned tags can be detected by testing, without killing the tag, if the kill password matches the original one stored in a back-end database. The risk of exposing the kill PIN of a given tag is however an important drawback of this approach.

Many signal-, power-, and blocking-based defenses, such as shielding of tags, use of noise, and third party guardians, can be found in the literature. The use of distance measurements to detect rogue readers has been discussed in [22]. Fishkin et al. propose the inclusion of low-cost circuitry in tags to use the signal-to-noise ratio of readers as a metric for trust. In [24], a similar assumption is used to determine if a reader is authorized to read the tag contents according to its physical distance. Castelluccia and Avoine propose in [10] the use of additional tags with better hardware capabilities than low-cost RFID hardware capabilities, to generate noise on the communication channel between readers and low-cost tags. The objective is to thwart possible eavesdroppers. Similar software-based blocking strategies can be found in [34,60]. Third party components with cryptographic features to perform authentication and acting as intermediaries between readers and tags have been proposed in [59,35]. The management of these components in real world scenarios like the supply chain of the retail industry is a problem and the main drawback of these proposals. Finally, the use of radio fingerprinting to detect characteristic properties of transmitted signals has also been considered in the literature. Cole and Ranasinghe [12] consider, however, that this technique is difficult to develop in RFID applications and that the benefits of using it, regarding performance, price and required implementation surface in tags, are unclear. Avoine and Oechslin discuss in [1] the prevention of traceability attacks via radio fingerprinting. They also conclude that obtaining radio fingerprints of tag is very expensive and difficult. The myriad of tags in circulation in future RFID scenarios would make impracticable the distinction of tags.

4.1 Towards Secret-Sharing Strategies

As an evolution of the minimalist cryptography approach presented by Juels in [30], and using lists of pseudonyms, the use of secret-sharing schemes is proposed by Langheinrich and Martin in [41,42] for solving authentication and privacy threats in low-cost RFID scenarios (e.g., supply chain applications of the retail industry). The work presented in [41] simplifies the lookup process performed from readers to back-end databases for identifying tags, while guaranteeing authentication and tracking resistance. Tag identifiers, seen in this work as the secrets, are encoded as a set of shares and stored in the internal memory of tags. The mechanism used by the authors to encode the shares is based on the *(t-n)*-threshold schemes of Shamir [63]. When the shares are cryptographically combined at the reader side, original tag identifiers are reconstructed. To prevent brute-force scanning from unauthorized readers — trying to obtain the complete set of shares — the authors propose a time-limited access that controls the amount of data sent from tags to readers. At the same time, a cache based process ensures that authorized readers quickly identify tags. Langheinrich and Martin extended the previous proposal to spread the set of shares across multiple tags [42]. Still based on the Shamir's secret sharing schemes, this approach encodes the indentifier of an item tagged with multiple RFID devices by distributing it into multiple shares stored within its tags. Authentication and privacy are enforced by requiring readers to obtain and combine the set of shares.

In [33], Juels, Pappu, and Parno present another secret-sharing based approach, but based on a dispersion of secrets strategy rather than an aggregation strategy — as used by Langheinrich and Marti in [41,42]. Two different schemes are discussed: dispersion of secrets across space and dispersion of secrets across time. In both schemes, a secret that is used to encrypt RFID identifiers (e.g., the EPC codes) is split in multiple shares and distributed among multiple parties. The construction and recombination of shares are based on the use of error-correcting codes. In order to identify a tag, a party must collect a number of shares. Privacy is achieved by the dispersion of secrets and encrypted identifiers. The dispersion approach helps to improve the authentication process between readers and tags, as tags move through a supply chain. Assuming that a given number of shares is necessary for a reader to obtain the EPC codes assigned to a pallet, for example, a situation where the number of shares obtained by a reader is not sufficient to reach the threshold leads to conclude that unauthorized tags are present on the pallet. The approaches presented in [33] increase the resistance of tags against unauthorized scanning by dispersing tag populations outside the supply chain. Without the space proximity to other tags with equivalent shares, an unauthorized reader cannot obtain the sufficient number of shares required to recover the original identifier of tags and items. A clear advantage of this approach is that it can be implemented on low-cost RFID tags, such as EPC Gen2 [20] tags, without requiring changes to the current specifications. Only an upgrade of readers is necessary. No real-world tests of the proposals have been conducted. The authors claim, although, that experiments for pharmaceutical products in a closed-loop supply chain are going to be conducted in the future. The main drawback of this approach is the amount of tag memory space required for storing the shares. A shrinking of key shares must be performed a priori in order to apply the scheme on current EPC tags. Other problems, such as tracking and information leaks due to the interaction between authorized readers and tags, must also be solved before deploying the schemes.

5 Conclusions

The constrained environment and threat model associated to low-cost RFID tags have stimulated the creation of a vast number of proposals to provide low-overhead security threat mitigation mechanisms in these devices. Vulnerable designs appeared in recent literature, such as the lightweight authentication protocols presented by Vajda and Buttyán in [68] (whose vulnerabilities were recently reported by Defend, Fu, and Juels in [15]), the set of ultra-lightweight authentication techniques presented by Peris-Lopez et al. in [54,55,56] (which were reported as vulnerable to passive [3] and active [44] attacks), and enhancements of these proposals, like the SASI protocol [11] (recently reported by Cao, Bertino, and Lei in [9] as vulnerable), show how challenging it is to design adequate procedures given the constraints. We surveyed lightweight defenses that can be useful to reduce the risk of threats. We addressed the methods according to three different perspectives: (1) one way hash-like defenses, (2) solutions relying on the single use of on-tag pseudorandomness and simple arithmetic operations; and (3) mechanisms not requiring the execution of cryptographic processes in the tags.

Regarding the first perspective, we pointed out the hardware challenges which to the best of our knowledge, are important obstacles for deployment in real world low-cost RFID scenarios like the supply chain of the retail industry. Physical One-Way Functions (POWFs) and Physically Unclonable Functions (PUFs) are a promising evolution of traditional hash-based protocols, but at a feasible production cost. Their sensitivity to physical noise, the large number of challenges and training session between readers and tags to guarantee adequate identification, and the difficulty to model and analyze, are open lines of research. In the second perspective, we pointed out the memory space and de-synchronization flaws as main limitations. The evolution of these solutions toward strategies that avoid the execution of on-tag cryptographic processes is heading recent researches. We pointed out the use of secret sharing strategies as a promising foundation for the management of keys for the design of authentication protocols and for dealing with privacy issues. Main drawbacks are the management of information leaks due to the interaction between readers and tags, and tracking of tags.

Acknowledgments. The authors graciously acknowledge the financial support received from the following organizations: Natural Sciences and Engineering Research Council of Canada (NSERC), Mathematics of Information Technology and Complex Systems (MITACS), Spanish Ministry of Science and Education (TSI2007-65406-C03-03 "E-AEGIS" and CONSOLIDER CSD2007-00004 "ARES" grants), and *La Caixa* (Canada awards).

References

1. Avoine, G., Oechslin, P.: RFID Traceability: A multilayer problem. In: S. Patrick, A., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 125–140. Springer, Heidelberg (2005)
2. Avoine, G., Oechslin, P.: A scalable and provably secure hash based RFID protocol. In: International Workshop on Pervasive Computing and Communication Security – PerSec 2005, pp. 110–114 (2005)
3. Barasz, M., Boros, B., Ligeti, P., Loja, K., Nagy, D.: Breaking LMAP. In: Conference on RFID Security, Malaga, Spain (July 2007)
4. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I.: An elliptic curve processor suitable for RFID-tags. Cryptology ePrint Archive, Report 2006/227 (2006)
5. Blackburn, S.R., Murphy, S., Paterson, K.G.: Comments on 'theory and applications of cellular automata in cryptography'. IEEE Trans. Softw. Eng. 23(9), 637–638 (1997)

6. Bolotnyy, L., Robins, G.: Physically unclonable function-based security and privacy in RFID systems. In: International Conference on Pervasive Computing and Communications – PerCom 2007, New York, USA, pp. 211–220. IEEE Press, Los Alamitos (2007)
7. Burmester, M., Van Le, T., de Medeiros, B.: Provably secure ubiquitous systems: Universally composable RFID authentication. In: 2nd International Conference on Security and Privacy in Communication Networks (SECURECOMM 2006). IEEE Press, Los Alamitos (2006)
8. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: IEEE Symp. On Foundations of Computer Science (FOCS 2001), pp. 136–145 (2001)
9. Cao, T., Bertino, E., Lei, H.: Security Analysis of the SASI Protocol. IEEE Transactions on Dependable and Secure Computing 6(1), 73–77 (2009)
10. Castelluccia, C., Avoine, G.: Noisy tags: a pretty good key exchange protocol for RFID tags. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 289–299. Springer, Heidelberg (2006)
11. Chien, H.: SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing 4(4), 337–340 (2007)
12. Cole, P., Ranasinghe, D. (eds.): Networked RFID Systems and Lightweight Cryptography — Raising Barriers to Product Counterfeiting, 1st edn. Springer, Heidelberg (2008)
13. Crawford, J.M., Kearns, M.J., Shapire, R.E.: The Minimal Disagreement Parity Problem as a Hard Satisfiability Problem. Tech. rep., Computational Intelligence Research Laboratory and AT&T Bell Labs (February 1994)
14. Daemen, J., Rijmen, V.: The Design of Rijndael: AES–the Advanced Encryption Standard. Springer, Heidelberg (2002)
15. Defend, B., Fu, K., Juels, A.: Cryptanalysis of two lightweight RFID authentication schemes. In: International Workshop on Pervasive Computing and Communication Security – PerSec 2007, New York, USA, pp. 211–216. IEEE Press, Los Alamitos (2007)
16. Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm, Athens, Greece. IEEE Press, Los Alamitos (2005)
17. Dolev, S., Kopeetsky, M.: Secure communication for RFIDs proactive information security within computational security. In: Datta, A.K., Gradinariu, M. (eds.) SSS 2006. LNCS, vol. 4280, pp. 290–303. Springer, Heidelberg (2006)
18. Dolev, S., Kopeetsky, M., Shamir, A.: RFID Authentication Efficient Proactive Information Security within Computational Security. Tech. rep., Department of Computer Science, Ben-Gurion University (July 2007)
19. Dolev, S., Kopeetsky, M., Clouser, T., Nesterenko, M.: Low Overhead RFID Security. In: Ahson, S.A., Ilyas, M. (eds.) RFID Handbook: Applications, Technology, Security, and Privacy, pp. 589–602, ch. 32. CRC Press, Boca Raton (2008)
20. EPCglobal. EPC Radio-frequency identity protocols Class-1 Generation-2. Technical report (January 2005), http://www.epcglobalinc.org/standards/
21. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 357–370. Springer, Heidelberg (2004)
22. Fishkin, K., Roy, S., Jiang, B.: Some methods for privacy in RFID communication. In: Castelluccia, C., Hartenstein, H., Paar, C., Westhoff, D. (eds.) ESAS 2004. LNCS, vol. 3313, pp. 42–53. Springer, Heidelberg (2005)
23. Gassend, B., Clarke, D., Dijk, M., Devadas, S.: Silicon physical random functions. In: 9th ACM conference on Computer and communications security, pp. 148–160. ACM, New York (2002)
24. Hancke, G.: Noisy carrier modulation for HF RFID. In: First International EURASIP Workshop on RFID Technology, Vienna, Austria (September 2007)

25. Henrici, D., Müller, P.: Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: International Workshop on Pervasive Computing and Communication Security – PerSec 2004, Orlando, Florida, USA, pp. 149–153. IEEE Computer Society, Los Alamitos (2004)
26. Holcomb, D., Burleson, W., Fu, K.: Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags. In: Third International Conference on RFID Security - RFIDSec 2007, Malaga, Spain (2007)
27. Hopper, N., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
28. ISO/IEC 18000-6:2004/amd:2006. Technical report (2006), http://www.iso.org/
29. Israsena, P.: Securing ubiquitous and low-cost RFID using tiny encryption algorithm. In: Intnl. Symp. on Wireless Pervasive Computing, Thailand. IEEE Press, Los Alamitos (2006)
30. Juels, A.: Minimalist cryptography for low-cost RFID tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)
31. Juels, A.: Strengthening EPC tags Against Cloning. In: WiSe 2005: Proceedings of the 4th ACM workshop on Wireless security, pp. 67–76. ACM Press, New York (2005)
32. Juels, A., Pappu, R.: Squealing euros: Privacy protection in RFID-enabled banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)
33. Juels, A., Pappu, R., Parno, B.: Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In: USENIX Security Symposium, San Jose, CA. USENIX (July-August 2008)
34. Juels, A., Rivest, R., Szydlo, M.: The blocker tag: Selective blocking of RFID tags for consumer privacy. In: 8th ACM Conf. Comput. Commun. Security, pp. 103–111 (2003)
35. Juels, A., Syverson, P., Bailey, D.: High-Power Proxies for Enhancing RFID Privacy and Utility. In: Danezis, G., Martin, D. (eds.) PET 2005. LNCS, vol. 3856, pp. 210–226. Springer, Heidelberg (2006)
36. Juels, A., Weis, S.: Authenticating pervasive devices with human protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
37. Juels, A., Weis, S.: Defining Strong Privacy for RFID. In: 5th Annual IEEE International Conference on Pervasive Computing and Communications, pp. 342–347. IEEE Press, Los Alamitos (2007)
38. Karthikeyan, S., Nesterenko, M.: RFID security without extensive cryptography. In: 3rd ACM workshop on Security of ad hoc and sensor networks, USA, pp. 63–67 (2005)
39. Katz, J., Shin, J.: Parallel and concurrent security of the HB and HB$^+$ protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer, Heidelberg (2006)
40. Kinosita, S., Hoshino, F., Komuro, T., Fujimura, A., Ohkubo, M.: Non-identifiable anonymous-ID scheme for RFID privacy protection (2003)
41. Langheinrich, M., Marti, R.: Practical Minimalist Cryptography for RFID Privacy. IEEE Systems Journal 1(2), 115–128 (2007)
42. Langheinrich, M., Marti, R.: RFID privacy using spatially distributed shared secrets. In: Ichikawa, H., Cho, W.-D., Satoh, I., Youn, H.Y. (eds.) UCS 2007. LNCS, vol. 4836, pp. 1–16. Springer, Heidelberg (2007)
43. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Symposium on Cryptography and Information Security (2006)
44. Li, T., Deng, R.: Vulnerability analysis of EMAP - an efficient RFID mutual authentication protocol. In: Second International Conference on Availability, Reliability and Security – AReS 2007, Vienna, Austria (April 2007)
45. Lim, D., Lee, J., Gassend, B., Suh, G., Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13(10), 1200–1205 (2005)

46. Mace, F., Standaert, F., Quisquater, J.: ASIC implementations of the block cipher sea for constrained applications. In: Conference on RFID Security, Spain, pp. 103–114 (2007)
47. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
48. Molnar, D., Wagner, D.: Privacy and security in library RFID: Issues, practices, and architectures. In: Conference on Computer and Communications Security – ACM CCS, Washington, DC, USA, pp. 210–219. ACM Press, New York (2004)
49. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient hash-chain based RFID privacy protection scheme. In: International Conference on Ubiquitous Computing – Ubicomp, Workshop Privacy: Current Status and Future Directions, Nottingham, England (September 2004)
50. Ouafi, K., Phan, R.: Privacy of Recent RFID Authentication Protocols. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 263–277. Springer, Heidelberg (2008)
51. Pappu, R.: Physical One-Way Functions. PhD thesis, MIT (2001)
52. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: LAMED, A PRNG for EPC Class-1 Generation-2 RFID specification. Computer Standards & Interfaces (2008)
53. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: An efficient authentication protocol for RFID systems resistant to active attacks. In: Denko, M.K., Shih, C.-s., Li, K.-C., Tsao, S.-L., Zeng, Q.-A., Park, S.H., Ko, Y.-B., Hung, S.-H., Park, J.-H. (eds.) EUC-WS 2007. LNCS, vol. 4809, pp. 781–794. Springer, Heidelberg (2007)
54. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In: Ecrypt, Austria (July 2006)
55. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 912–923. Springer, Heidelberg (2006)
56. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 352–361. Springer, Heidelberg (2006)
57. Piramuthu, S.: HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In: Collaborative Electronic Commerce Technology and Research – CollECTeR 2006, Basel, Switzerland (June 2006)
58. Ranasinghe, D., Engels, D., Cole, P.: Low-cost RFID systems: Confronting security and privacy. In: Auto-ID Labs Research Workshop, Zurich, Switzerland (September 2004)
59. Rieback, M., Crispo, B., Tanenbaum, A.: RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 184–194. Springer, Heidelberg (2005)
60. Rieback, M., Crispo, B., Tanenbaum, A.: Keep on blockin'in the free world: Personal access control for low-cost RFID tags. In: 13th Cambridge Workshop on Security Protocols. Springer, Heidelberg (2005)
61. Roussos, G.: Enabling RFID in retail. IEEE Computer 39(3), 25–30 (2006)
62. Sarma, S.: Toward the 5 cent tag. White Paper, Auto-ID Center (November 2001)
63. Shamir, A.: How to share a secret. Commun. of the ACM 22(11), 612–613 (1979)
64. Skoric, B., Tuyls, P.: Secret key generation from classical physics. Philips Research Book Series (September 2005)
65. Takaragi, K., Usami, M., Imura, R., Itsuki, R., Satoh, T.: An ultra small individual recognition security chip. IEEE Micro. 21(6), 43–49 (2001)

66. Tsudik, G.: YA-TRAP: Yet another trivial RFID authentication protocol. In: International Conference on Pervasive Computing and Communications – PerCom 2006, Pisa, Italy. IEEE Press, Los Alamitos (2006)
67. Tuyls, P., Batina, L.: RFID-tags for anti-counterfeiting. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 115–131. Springer, Heidelberg (2006)
68. Vajda, I., Buttyán, L.: Lightweight authentication protocols for low-cost RFID tags. In: Second Workshop on Security in Ubiquitous Computing, Seattle, WA, USA (2003)
69. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)
70. Wheeler, D., Needham, R.: TEA, a Tiny Encryption Algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1995)
71. Wolfram, S.: A new kind of science. Wolfram Media Inc., Champaign (2002)
72. Wolkerstorfer, J.: Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags? In: Handout of the Ecrypt Workshop on RFID and Lightweight Crypto (July 2005)

An Effective TCP/IP Fingerprinting Technique Based on Strange Attractors Classification

João Paulo S. Medeiros, Agostinho M. Brito Jr., and Paulo S. Motta Pires

LabSIN - Security Information Laboratory
Department of Computer Engineering and Automation – DCA
Federal University of Rio Grande do Norte – UFRN
Natal, 59.078-970, RN, Brazil
{joaomedeiros,ambj,pmotta}@dca.ufrn.br

Abstract. We propose a new technique to perform TCP/IP (Transmission Control Protocol/Internet Protocol) stack fingerprinting. Our technique relies on chaotic dynamics theory and artificial neural networks applied to TCP ISN (Initial Sequence Number) samples making possible to associate strange attractors to operating systems. We show that it is possible to recognize operating systems using only an open TCP port on the target machine. Also, we present results which shows that our technique cannot be fooled by Honeyd or affected by PAT (Port Address Translation) environments.

1 Introduction

Operating System (OS) fingerprinting is the process of identifying the operating systems of a machine through a computer network. TCP/IP stack fingerprinting is a technique that uses distinguishable characteristics of operating systems TCP/IP protocols implementations to perform OS fingerprinting. The components and subprocess of OS fingerprinting are presented in Figure 1 [1].

As shown in Figure 1, the overall process starts with acquisition of network data related to the target machine. After that, an algorithm is used to create a signature (fingerprint) which represents the target OS. This signature is then compared, using a matching algorithm, with known OSes signatures previously

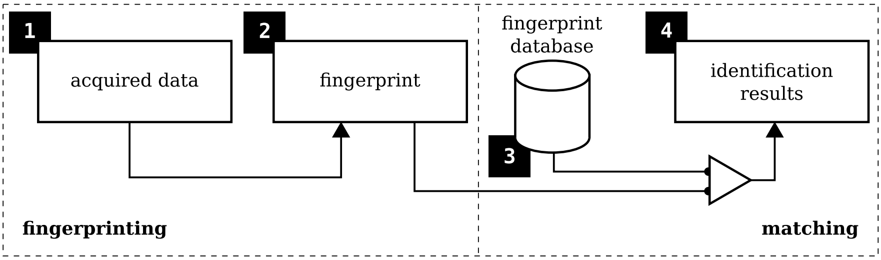


Fig. 1. Graphical representation of OS fingerprinting process

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 208–221, 2010.

stored on a fingerprint database. Finally, the identification result for the target OS is presented.

The process of discovering the operating system of a networked machine is an initial requirement for a large range of network security tasks. Among these tasks, we can highlight the identification of vulnerable hosts, aid exploiting process and building up network inventory [2]. However, the use of Port Address Translation (PAT) and tools like Honeyd [3] makes TCP/IP fingerprinting a difficult task [4]. Techniques and tools like those ones affect the data collected by the OS fingerprinting process. As an example of this statement, SinFP [5] fingerprint tool works on address translation environments, but it is inefficient when the target is a machine created using Honeyd.

Michael Zalewski made an analysis of TCP ISN generators for a large set of OSes [6]. In his work, strange attractors built from TCP ISN samples of several operating systems are used to predict the next TCP ISN. It is a known fact that these attractors can be used to perform OS fingerprinting, although, until now to our best knowledge, there is not any tool that perform OS fingerprinting classifying these attractors [7]. An attractor represents states of a dynamical system and expresses how this system evolves. The attractor is plotted in a phase space. A phase space is a mathematical space where each coordinate represents a variable that compounds one state of a dynamical system [8]. Each point in the phase space represents one possible state of the system [9].

To build an attractor based only on a function $s(t)$ that represents the output of a dynamical system at time t, we use a method named *delay coordinates*. Each point $\mathbf{x}$ with coordinates $[x_1, x_2, x_3, \ldots, x_m]$ in the attractor is defined as

$$[x_1, x_2, x_3, \ldots, x_m] = [s(t), s(t-\tau), s(t-2\tau), \ldots, s(t-(m-1)\tau)] \qquad (1)$$

where τ is a time delay applied to the series $s(t)$ and m is the embedding dimension of the attractor [10]. Successive applications of Equation 1 over time t creates a trajectory in a m-dimensional space.

We propose a new technique based on the chaotic dynamic theory and artificial neural networks to perform TCP ISN stack fingerprinting to identify operating systems. Our work presents a new TCP/IP stack fingerprinting technique which cannot be fooled by Honeyd and that is able to perform OS fingerprinting even on PAT environments. It is important to note that other tools use TCP ISN to perform OS fingerprinting, but their fingerprint component (the second one in Figure 1) is simple and may be deceived [2,5].

The rest of the paper is organized as follows. In the Section 2, we present the state of art of TCP ISN generators and how to create attractors from them. Also, in this Section, we present the testbed used to acquire data and to perform the tests. To use attractors as a basis to perform OS fingerprinting we need to produce comparable representations of them. We use the Kohonen Self-Organizing Maps (SOM) [11] to build up these representations. Using the resulting set of points achieved by neural network training, and after some post-processing, we can build a three-dimensional set of oriented points that represents the properties of an attractor. In Section 3, we present our method to represent and classify attractors. The proposed method is validated by the results presented in Section 4. We conclude our paper with Section 5, which shows the significance of our results for TCP/IP fingerprinting, and making some recommendations for future research direction.

2 TCP Initial Sequence Number

The TCP ISN is used in a TCP communication to avoid duplicated segments from previous incarnations of the connection [12]. The way these numbers are generated leads to security flaws according to their predictability [13]. A recommendation proposed by RFC 1948 [13] was released since vulnerabilities were associated to the first TCP ISN implementations. From the CERT Advisory CA-2001-09 [14] we can extract the current TCP ISN generation recommendation

$$G_{isn}(t) = M(t) + F(\cdot) \tag{2}$$

$$M(t) = M(t-1) + R(t) \tag{3}$$

$$F(\cdot) = f(connection\_id, secret\_key) \tag{4}$$

where $G_{isn}(t)$ is the TCP ISN generator, $M(t)$ is a random incremental function and $F(\cdot)$ is a connection dependent term that is constant most of the time. The first argument of $F(\cdot)$, *connection_id*, is composed by the source and target IP addresses and ports. The second argument of $F(\cdot)$, *secret_key*, is an optional input that may change over time or is a constant unknown value. In our work we used the PRNG (Pseudo Random Number Generator) $R(t)$ function to build attractors. We can obtain the values of this function using samples of ISN values, as expressed in Equation 2. Using Equations 2, 3 and 4, we might express $R(t)$ as an estimated function

$$\hat{R}(t) = G_{isn}(t) - G_{isn}(t-1). \tag{5}$$

When $F(\cdot)$ changes, the estimated value $\hat{R}(t)$ is different of $R(t)$. This can be viewed as a noise that appears with a small frequency. Another noise associated with $\hat{R}(t)$ occurs when the increment of $M(t)$ exceeds the 32 bits limit of the TCP sequence number field. The TCP ISN generators that do not seem to follow RFC 1948 have their PRNG term presented by $G_{isn}(t)$ instead of the $\hat{R}(t)$.

In our work we acquired TCP ISN data samples from the following OSes: FreeBSD 7.0, Cisco IOS 12.3.11, Slackware 12.1 (Linux 2.6.24.5), NetBSD 4.0, OpenBSD 4.3, QNX 6.3.0, Sun Solaris 10, Microsoft Windows Vista Ultimate and Microsoft Windows XP Pro Service Pack 2. We have also collected TCP ISN samples of Honeyd 1.5c [3].

The data to perform all the tests has been obtained using the testbed presented in Figure 2. The `fingerprinter` machine runs a Slackware Linux operating system machine and has been used to acquire TCP ISN samples as well as to conduct tests with our TCP/IP fingerprinting tool. All analysed OSes were installed on the `target systems` machines. On the `honeyd and pf` machine,

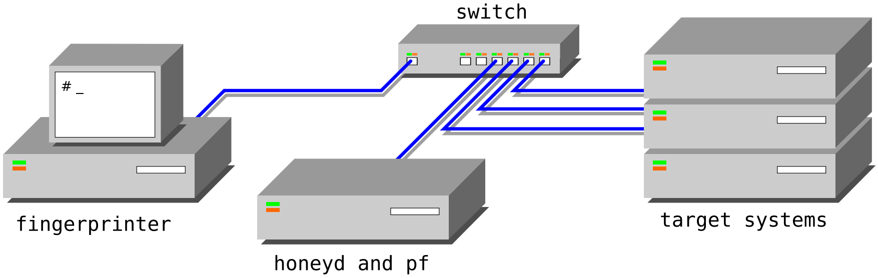


Fig. 2. Testbed used to acquire data and to perform tests

an OpenBSD 4.3 operating system machine has been configured to run Honeyd and PAT. The PAT setup has been implemented using the OpenBSD Packet Filter [15]. The PAT environment was used to confirm that both address translation and packet normalization do not affect our tool. Our tool sends SYN packets to other machines, capture the SYN ACK response packets and send RST packets to avoid SYN flood blocking.

For each OS we have captured 100000 TCP ISN samples, and other 10000 samples were taken by varying IP addresses, TCP ports, clock time and system boot time, to verify $\hat{R}(t)$ independence on these parameters as suggested by Equation 5. These numbers were chosen because they have been considered, after some empirical evaluations, large enough to conduct our tests. During the tests, we have found that a fewer number of packets is enough to identify each OS. By Equation 1, each point of the attractor is composed by a set of values of one function with delays multiple of τ. In our case, each SYN packet is sent 10ms after the previous one ($\tau = 10^{-2}$s). This time interval has been chosen because we have noted that some operating systems, for example FreeBSD, keeps the same TCP ISN value for a long time if the τ value is smaller than 10ms.

Equation 1 was used to create each point $\mathbf{x}$ of the attractor embedded in a bi-dimensional phase space for a given OS as follows

$$[x_1, x_2] = [\hat{R}(t), \hat{R}(t-1)]. \tag{6}$$

In Figure 3, we presented the attractor of each OS [16]. All data presented in graphics are normalized, so we suppressed the axis values. Attractors (d) and (f) show a slight difference in scale. This happens because of the presence of a noisy value in (f) (pointed by two arrows). Another example of noisy data is show in attractor (c). Because of the 32 limit of TCP sequence number field, the value of $\hat{R}(t)$ is too large when the TCP ISN value wrap around. It is important to note that these noises make these attractors a problematic representation to be used in OS fingerprinting. The use of SOM neural network intends to minimize the effect of these noisy values.

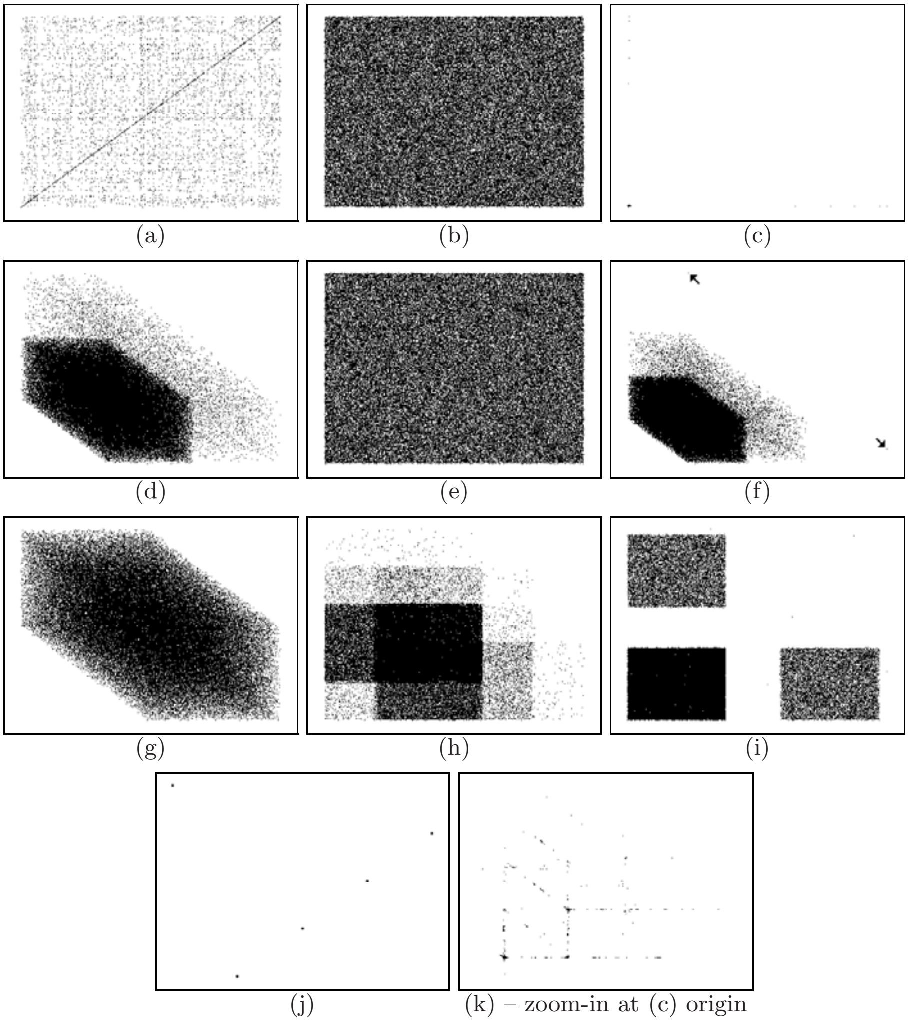

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) – zoom-in at (c) origin

Fig. 3. Attractors of the analyzed operating systems. (a) FreeBSD 7.0; (b) Cisco IOS 12.3.11; (c) Slackware 12.1 (Linux 2.6.24.5); (d) NetBSD 4.0; (e) OpenBSD 4.3; (f) QNX 6.3.0; (g) Sun Solaris 10; (h) Microsoft Windows Vista Ultimate; (i) Microsoft Windows XP Pro Service Pack 2, and (j) Honeyd 1.5c.

3 Classification

To perform OS fingerprinting, we need to create a signature for each target OS, by building a signature database that will serve as comparison basis to signatures. This signature database can be created through the characterization of strange attractors, which in our work is performed by a SOM neural network [11]. The use of this kind of neural network to characterize attractors [17] and to classify OS fingerprint is well referenced [18] [19].

We use a SOM generated map to express the shape of attractors because it can minimize the changing effects of $F(\cdot)$ and the 32 bit limit of TCP ISN field. At first, we apply a post-processing technique based on a sum of Gaussian functions to express the density of each region of the attractor. Then we use a weighted sum of directions to specify each attractor point's direction. Finally, the attractor characterization is done, and they are ready to be used to classify new attractors. To perform this task we need to setup a metric that measures the distance between two signatures.

3.1 Self-Organizing Maps

To represent the shape of the attractors, it is necessary to use an algorithm able to provide shape description. The choice of the appropriate algorithm plays a fundamental role in our process due to the presence of noisy data in the function $\hat{R}(t)$, as discussed in the previous Section. The SOM learning algorithm is able to find the main statistical properties of input space. Using SOM, the noisy data that is a small part of input will be ignored if it does not follows the same statistics of input.

The classical SOM neural network is composed by two basic elements: an output layer, that consists in a $M \times N$ regular grid of neurons with a set of d-dimensional weights, and an d-dimensional input vector. The architecture of SOM and its elements are presented in Figure 4 [11].

In the SOM training algorithm, the samples of the input space are presented to the network. For each input sample the algorithm computes the winning neuron, that is the closest neuron in output layer (competitive process), and adjusts its weights and the weights of the neighborhood neurons (cooperative process). In the beginning, the neighborhood criteria is large enough to adjust the weights of all neurons, and the learning rate is also high. On the first steps of training

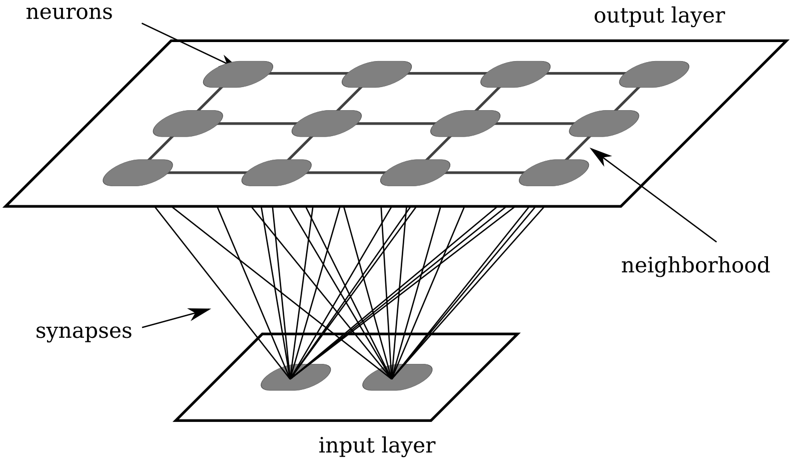


Fig. 4. SOM neural network structure

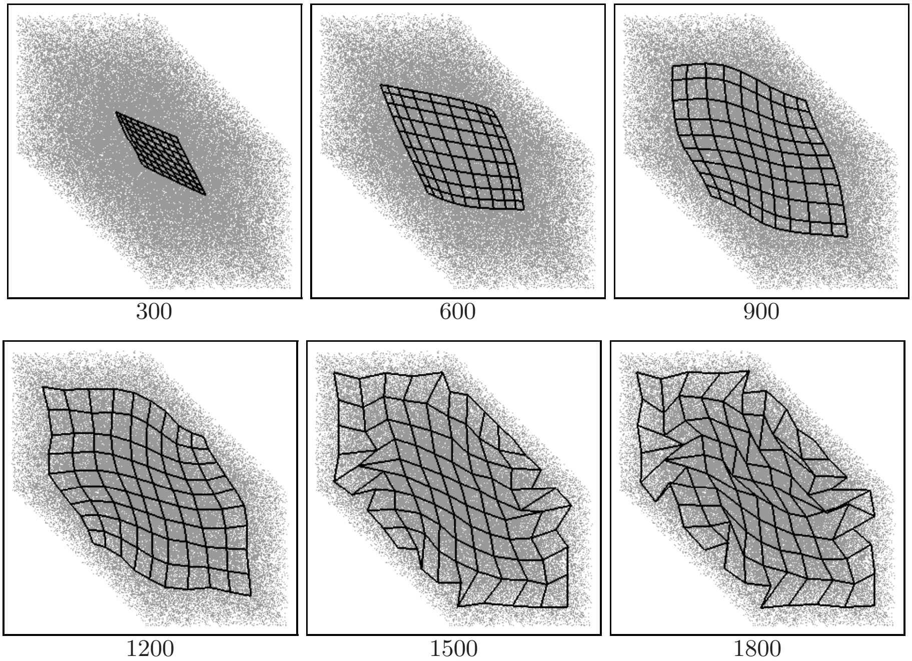


Fig. 5. Neural network training evolution

algorithm, the neurons are kept in order according to the input space. These steps are part of the *ordering phase*. When the neighborhood criteria and the learning rate decreases enough, the output neurons make a fine adjust and the algorithm is in the *convergence phase*.

Some iterations plots are presented in Figure 5. The first three plots refer to the *ordering phase* and the others refer to *convergence phase*. The process described in the last paragraph has many applications, such as 3D reconstitution, non-linear Principal Component Analysis (PCA), data mining and non-linear quantization.

We have applied the SOM neural network to build a representation for the attractor's shape. The SOM neural network creates a lattice of connected points whose geometry can be used to build a description of the shape. Figure 6 (a) shows the set of points generated by the attractor for Sun Solaris 10 in Figure 3 (g).

The size of this set is 900 (30x30 SOM topology) and it is generated after 1800 training epochs of the conventional SOM training algorithm. All the generated points are normalized after the training. To simplify our formulations we call P the generated set of points. After SOM training just the attractor's shape is represented. The attractor density is already represented by points distances in Figure 6 (a). However, we reinforce this information by using the concept of density factor $\phi(\mathbf{p})$ for a point $\mathbf{p} \in P$

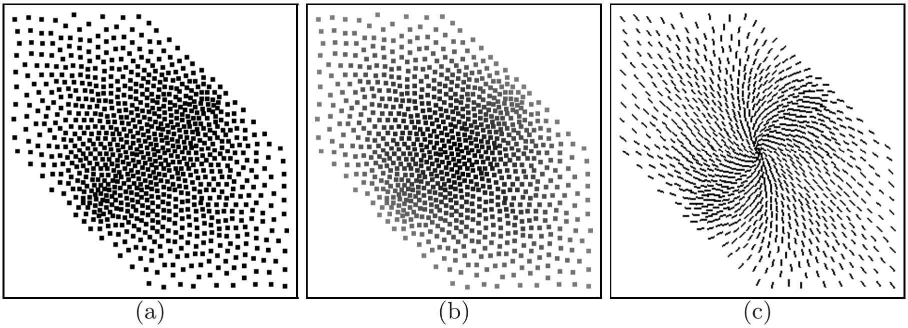


Fig. 6. Attractor's representations for Sun Solaris 10

$$\phi(\mathbf{p}) = \sum_{i=1}^{N} \exp\left(-\frac{\|\mathbf{p}-\mathbf{x}_i\|^2}{2\sigma_d^2}\right) \tag{7}$$

where $\mathbf{x}_i$ represents the i-th element of the input set X; N is the size of the input set, and σ_d is the Gaussian width parameter. Each point in P has its density factor normalized after the application of Equation 7. Figure 6 (b) shows the result for Sun Solaris 10, and Figure 7 illustrates the results for the other systems used in this work.

The density factor was mapped to point color, from gray to black, to make visualization easier. We used on tests $\sigma_d = 0.05$, but, in general, this parameter depends on the number of input samples.

To map the attractor's flow we create an orientation $\theta(\mathbf{p})$ for each point $\mathbf{p} \in P$. To include this information, we consider the direction of each point in the original data set according to its distance to $\mathbf{p}$

$$\theta(\mathbf{p}) = \tan^{-1}\left[\sum_{i=1}^{N-1} \exp\left(-\frac{\|\mathbf{p}-\mathbf{x}_i\|^2}{2\sigma_o^2}\right)(\mathbf{x}_{i+1}-\mathbf{x}_i)\right] \tag{8}$$

where $\mathbf{x}_i$ represents the i-th element of the input set X; N is the size of the input set, and σ_o is the Gaussian width parameter. We used $\sigma_o = 0.05$. Figure 6 (c) illustrates the result for Sun Solaris 10, and Figure 8 illustrates the results for the other operating systems.

Comparing attractors in Figure 7 (d) and (f), we can see that the SOM algorithm removed the noisy data effect present in Figure 3 (f). As demonstrated in this situation, the use of SOM algorithm decreased the number of points used to represent attractors and provides a statistical-based mechanism to reduce the effects of noisy data.

To perform OS fingerprinting we must be able to compare two signatures and figure out their similarities. Next subsection presents the metric used to compare two attractor representations.

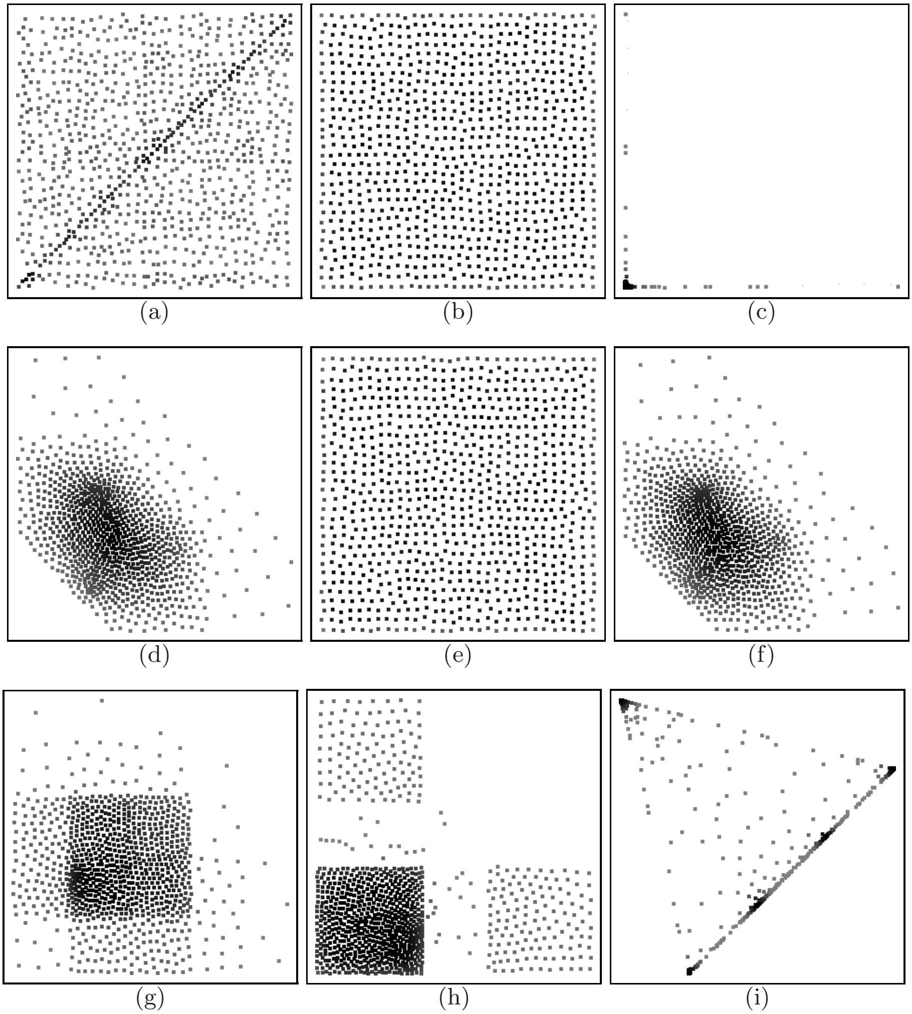


Fig. 7. Attractors density. (a) FreeBSD 7.0; (b) Cisco IOS 12.3.11; (c) Slackware 12.1 (Linux 2.6.24.5); (d) NetBSD 4.0; (e) OpenBSD 4.3; (f) QNX 6.3.0; (g) Microsoft Windows Vista Ultimate; (h) Microsoft Windows XP Pro Service Pack 2, and (i) Honeyd 1.5c.

3.2 Hausdorff Based Matching

In this Section, an attractor, A, is represented as a 3-tuple $\langle P,\varPhi,\varTheta\rangle$ where P is the set of positions in a bi-dimensional space, each position represented by $\mathbf{p}$; $\varPhi$ is the set of density factors given by $\phi(\mathbf{p})$; and $\varTheta$ is the set of orientations given by $\theta(\mathbf{p})$. Each element a of the atractor, $a \in A$, is expressed using the notation $\langle \mathbf{p}, \phi, \theta\rangle$. In this sense, an element a has its position represented by $\mathbf{p}_a$, its density represented by ϕ_a, and its orientation represented by θ_a.

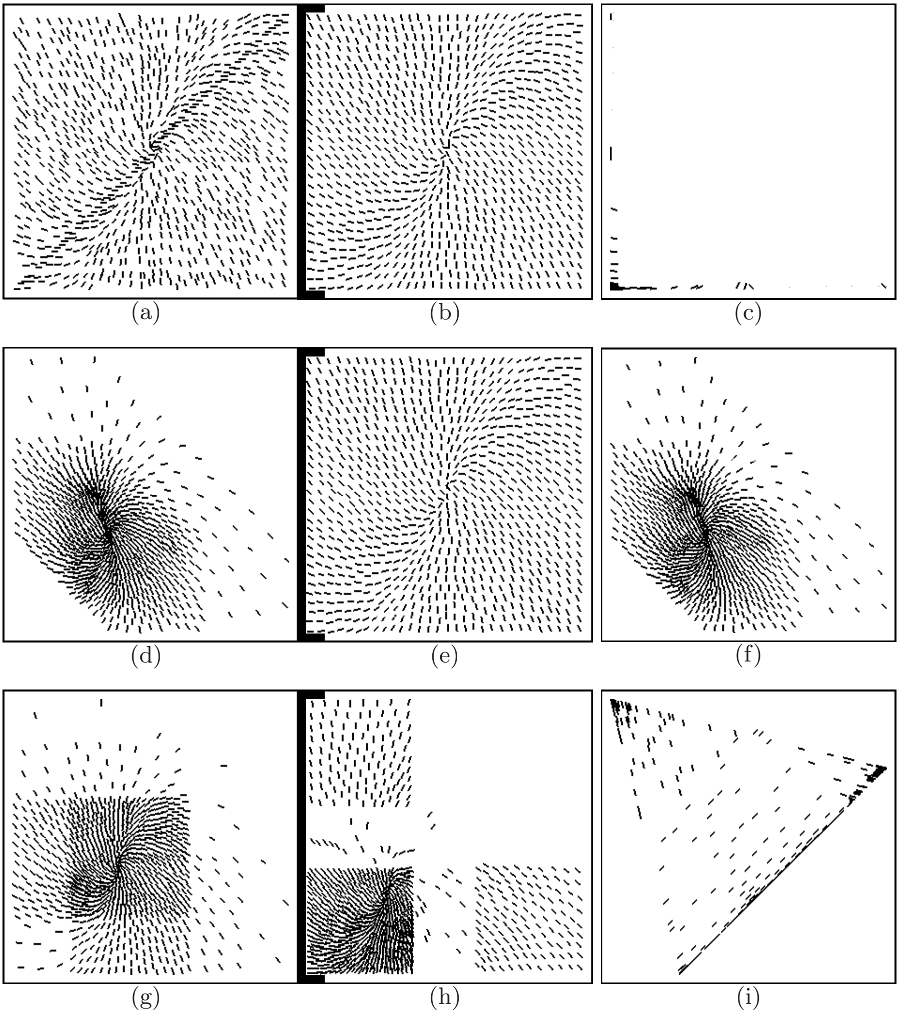


Fig. 8. Strange attractors flow. (a) FreeBSD 7.0; (b) Cisco IOS 12.3.11; (c) Slackware 12.1 (Linux 2.6.24.5); (d) NetBSD 4.0; (e) OpenBSD 4.3; (f) QNX 6.3.0; (g) Microsoft Windows Vista Ultimate; (h) Microsoft Windows XP Pro Service Pack 2, and (i) Honeyd 1.5c.

To compute the dissimilarity between two representations of attractors we propose two metrics based on the Hausdorff distance [20]. Given two attractor representations X and Y, the first metric $N(X, Y, \alpha)$ consists on counting the points $x \in X$ whose direction differs at least from a given parameter α to the closest point $y \in Y$

$$N(X, Y, \alpha) = \left| \{x \in X : [x \cdot \arg \inf_{y \in Y} d_N(x, y)] \leq \alpha\} \right| \tag{9}$$

$$d_N(x, y) = \|\langle \mathbf{p}_x, \phi_x \rangle - \langle \mathbf{p}_y, \phi_y \rangle\| \tag{10}$$

where $|\cdot|$ represents the cardinality of the set defined by the right hand side of the Equation 9, $\langle \mathbf{p}_x, \phi_x \rangle$ and $\langle \mathbf{p}_y, \phi_y \rangle$ represents vectors composed by the position $\mathbf{p}$ and the density ϕ of the element x and y, respectively, and $\| \cdot \|$ represents the Euclidean distance between the vectors $\langle \mathbf{p}_x, \phi_x \rangle$ and $\langle \mathbf{p}_y, \phi_y \rangle$. The dot product in the Equation 9 is performed using the directions of the elements.

The second metric, also based on Hausdorff distance, returns the number of elements $x \in X$ that is nearest to $y \in Y$ given a parameter β, as shown by the following equations

$$H(X, Y, \beta) = \left| \{x \in X : \inf_{y \in Y} d_H(x, y) \geq \beta\} \right| \tag{11}$$

$$d_H(x, y) = \|\mathbf{p}_x - \mathbf{p}_y\| \tag{12}$$

Since the metrics N and H are not symmetric, i.e. $N(X, Y, \alpha) \neq N(Y, X, \alpha)$ and $H(X, Y, \beta) \neq H(Y, X, \beta)$, we defined two auxiliary symmetric metrics

$$N_s(X, Y, \alpha) = \max\{N(X, Y, \alpha), N(Y, X, \alpha)\} \tag{13}$$

$$H_s(X, Y, \beta) = \max\{H(X, Y, \beta), H(Y, X, \beta)\} \tag{14}$$

These metrics will act as a matching algorithm on the proposed OS fingerprinting process. The symmetric metrics are used in next section to classify attractor representations.

4 Results

We used a 30x30 Kohonen neural network to build each attractor representation. The number of training epochs was fixed in 1100 and the size of the input set, used to build each OS fingerprint signature, was 100000. We used other 10000 TCP ISN samples of each OS to create attractor representations to validate our technique. The parameters $\alpha = 0.1$ and $\beta = 0.3$, used in the metrics equations, were chosen empirically to maximizes the absolute value between of the correct and the incorrect classification results. Using Equations 13 and 14, the symmetric metrics for operating systems attractors pairs, A and B, are computed using 10000 and 100000 TCP ISN samples, respectively.

Table 1 shows the values of the symmetric metrics $H(X, Y, \beta)$ and $N_s(X, Y, \alpha)$, Equations 13 and 14. We have used A to represent attractors built using 10000 TCP ISN samples and b to represent attractors for built using 100000 TCP ISN samples. We compute the symmetric metrics attractor fingerprint signature for each OS used in this paper. As seen on Table 1, our method classify the attractor correctly in all cases.

The values obtained for OpenBSD and Cisco IOS tests show a close relation between these OSes. This fact can also be verified for NetBSD and QNX, but the use of NetBSD code in QNX is a known fact [21]. So, we can assume in both cases, based in our results, that they use the same TCP ISN generator.

Table 1 shows that each metric is more suitable for some OSes. For example, the metric $H_s(X, Y, \beta)$ is more suitable to classify NetBSD, QNX, Solaris, and Windows Vista Ultimate, whereas the metric $N_s(X, Y, \alpha)$ is more suitable to classify Slackware (Linux), IOS e OpenBSD. This fact suggest that the use of an

Table 1. Classification of TCP ISN sets. The "$*$" indicates the smaller value for the N_s and H_s metrics.

A	B	$H_s(A, B, \beta)$	$N_s(A, B, \alpha)$	A	B	$H_s(A, B, \beta)$	$N_s(A, B, \alpha)$
FreeBSD	FreeBSD	*62	*121	OpenBSD	FreeBSD	351	275
	Honeyd	627	615		Honeyd	629	638
	Cisco IOS	680	376		Cisco IOS	113	67
	Linux	883	447		Linux	893	510
	NetBSD	545	413		NetBSD	525	405
	OpenBSD	657	282		OpenBSD	*78	*54
	QNX	567	442		QNX	521	407
	Solaris	375	383		Solaris	383	186
	Windows Vista	501	470		Windows Vista	418	204
	Windows XP	622	579		Windows XP	627	382
Honeyd	FreeBSD	621	634	QNX	FreeBSD	381	538
	Honeyd	*0	*8		Honeyd	719	775
	Cisco IOS	662	647		Cisco IOS	585	431
	Linux	891	894		Linux	894	708
	NetBSD	581	755		NetBSD	*0	*18
	OpenBSD	642	612		OpenBSD	601	434
	QNX	577	729		QNX	*0	33
	Solaris	677	705		Solaris	469	474
	Windows Vista	594	697		Windows Vista	286	258
	Windows XP	600	729		Windows XP	477	332
Cisco IOS	FreeBSD	182	154	Solaris	FreeBSD	258	256
	Honeyd	724	620		Honeyd	701	738
	Cisco IOS	181	77		Cisco IOS	551	190
	Linux	889	479		Linux	892	555
	NetBSD	454	435		NetBSD	451	447
	OpenBSD	*141	*66		OpenBSD	541	162
	QNX	470	460		QNX	475	477
	Solaris	282	164		Solaris	*0	*21
	Windows Vista	315	249		Windows Vista	140	167
	Windows XP	593	545		Windows XP	656	524
Linux	FreeBSD	651	547	Windows Vista	FreeBSD	359	501
	Honeyd	786	886		Honeyd	609	772
	Cisco IOS	854	636		Cisco IOS	375	104
	Linux	*90	*50		Linux	887	680
	NetBSD	805	684		NetBSD	413	423
	OpenBSD	848	591		OpenBSD	422	139
	QNX	810	709		QNX	413	451
	Solaris	864	847		Solaris	54	120
	Windows Vista	860	695		Windows Vista	*7	*96
	Windows XP	629	559		Windows XP	604	557
NetBSD	FreeBSD	358	542	Windows XP	FreeBSD	459	602
	Honeyd	583	764		Honeyd	652	727
	Cisco IOS	572	401		Cisco IOS	688	520
	Linux	895	693		Linux	738	642
	NetBSD	*0	*16		NetBSD	474	325
	OpenBSD	582	397		OpenBSD	676	491
	QNX	*0	36		QNX	451	319
	Solaris	455	441		Solaris	670	573
	Windows Vista	263	211		Windows Vista	585	536
	Windows XP	479	339		Windows XP	*1	*40

hybrid solution will produce more efficient results in terms of number of samples required to identify OSes.

5 Conclusions

In this paper, we propose a new technique based on chaotic dynamics theory and neural networks to identify operating systems. This technique performs TCP/IP stack fingerprinting using TCP ISN samples. We presented results which shows that our technique cannot be fooled by Honeyd or affected by PAT environments. Our technique can be used to build compact representations for strange attractors, classify dynamical systems and network services that use PRNG in their protocols. The technique can also be used to group OSes that have the same TCP ISN generator. The operating system identification process in these cases can be made using other procedures.

Future works include the analysis of other operating systems and the implementation of optimization techniques for parameter setup to replace empirical value settings used in this work.

We are developing a tool, supported by Google Summer of Code 2009 program, as an Umit Project application. We intend to test this tool against remote hosts through the Internet to analyse the Round-Trip Time (RTT) effect on TCP ISN sample acquisition.

Acknowledgment

The authors would like to express their gratitude to the Department of Computer Engineering and Automation from Federal University of Rio Grande do Norte, Brazil, and REDIC (Instrumentation and Control Research Network) for the support received along the development of this work.

References

1. Medeiros, J.P.S., Brito Jr., A.M., Pires, P.S.M.: A Data Mining Based Analysis of Nmap Operating System Fingerprint Database. In: Proceedings of the 2nd International Workshop on Computational Intelligence in Security for Information Systems (CISIS 2009). Advances in Intelligent and Soft Computing, vol. 63 (to be published, 2009)
2. Fyodor: Nmap (2009), http://www.nmap.org/
3. Provos, N.: Honeyd (2008), http://www.honeyd.org/
4. Provos, N., Holz, T.: Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley, Reading (2008)
5. Auffret, P.: SinFP (2008), http://www.gomor.org/bin/view/Sinfp
6. Zalewski, M.: Strange attractors and TCP/IP sequence number analysis (2001), http://lcamtuf.coredump.cx/oldtcp/tcpseq.html
7. Veysset, F., Courtay, O., Heen, O., et al.: New tool and technique for remote operating system fingerprinting. Intranode Software Technologies (2002)

8. Baker, G.L., Gollub, J.P.: Chaotic Dynamics: An Introduction, 2nd edn. Cambridge University Press, Cambridge (1996)
9. Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge University Press, Cambridge (2002)
10. Alligood, K., Sauer, T., Yorke, J.: Chaos: an introduction to dynamical systems. Springer, Heidelberg (1997)
11. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
12. Postel, J.: RFC 793: Transmission control protocol. Technical report (1996)
13. Bellovin, S.: RFC 1948: Defending Against Sequence Number Attacks. Technical report (1996)
14. CERT: CERT advisory CA-2001-09 statistical weaknesses in TCP/IP initial sequence numbers (2001), http://www.cert.org/advisories/CA-2001-09.html
15. OpenBSD: PF: The OpenBSD Packet Filter (2008), http://www.openbsd.org/faq/pf/
16. Medeiros, J.P.S., Brito Jr., A.M., Pires, P.S.M.: A new method for recognizing operating systems of automation devices. In: Proc. IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2009 (to be published, 2009)
17. Goerke, N., Kintzler, F., Eckmiller, R.: Self organized classification of chaotic domains from a nonlinear attractor. In: Proc. International Joint Conference on Neural Networks (IJCNN 2001), Washington, DC, July 2001, vol. 3 (2001)
18. Medeiros, J.P.S., Cunha, A.C., Brito Jr., A.M., Pires, P.S.M.: Application of Kohonen maps to improve security tests on automation devices. In: Lopez, J., Hämmerli, B.M. (eds.) CRITIS 2007. LNCS, vol. 5141, Springer, Heidelberg (2008)
19. Medeiros, J.P.S., Cunha, A.C., Brito Jr., A.M., Pires, P.S.M.: Automating security tests for industrial automation devices using neural networks. In: Proc. IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007), pp. 772–775 (2007)
20. Deza, E., Deza, M.M.: Dictionary of Distances. Elsevier Science, Amsterdam (2006)
21. NetBSD Project: Products based on NetBSD (2009), http://www.netbsd.org/gallery/products.html

DDoS Defense Mechanisms: A New Taxonomy

Astha Keshariya and Noria Foukia

Department of Information science
University of Otago, New Zealand

Abstract. Ever expanding array of schemes for detection and prevention of Distributed Denial of Service (DDoS) attacks demands for a constant review and their categorization. As detection techniques have existed for a relatively longer period of time than defense mechanisms, researchers have categorized almost all the existing and expected forthcoming attacks. However, techniques for defense are still nurturing. Researchers have explored that there could be diverse ways of launching DDoS attacks. Consequently, need of defense strategy that adapts and responds autonomously to these variety of attacks is imperative. As more and more excavation is done in the arena of DDoS Defense Mechanisms, we understand that along with the conventional, well known DDoS Prevention and mitigation mechanism there are other factors that play equally important role in shielding a system from DDoS attacks. Deployment strategy, degree of cooperation of the internet host, code of behaviour while the system is already under attack, and post-attack analysis, etc, are such factors. In this paper, we have assorted the existing enormous defense mechanisms, and proposed an enhanced taxonomy that incorporates possible parameters that might influence DDoS Defense.

Keywords: Distributed Denial of Service, Taxonomy, Autonomous Defense mechanisms.

1 Introduction

A Denial-of-Service (DoS) attack is widely regarded as a major threat to the Internet since it causes loss of service and network connectivity to legitimate users or the entire network, by consuming its bandwidth and exhausting its resources. When multiple (compromised) hosts act in a coordinated fashion, it is referred as Distributed Denial-of-Service (DDoS). Eventually, DDoS traffic creates a heavy congestion in the Internet core and interrupts all Internet users whose packets cross congested routers.

Traditionally, DDoS attack is a sequential process comprised of: 1) scanning for vulnerable remote machines 2) discovered vulnerability is then exploited to break into recruited machines (called agents) and infect them (with the attack code). Attacker may also distribute attack code camouflaging itself as a useful application (called Trojans). Generally, the attack code has the potential to propagate itself and infect other agents, once installed. 3) the handler (called Master) orchestrates these unaware compromised hosts (called Zombie) while

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 222–236, 2010.

hiding identity of agents during the attack (by IP spoofing). This, ignorant, army of zombies then attacks the victim from an assortment of attacks such as flooding attack[1], reflector attack[2] that exists in the wild.

2 DDoS Defense Overview

The ever increasing sophistication of DDoS attackers and attack tools is causing detection and protection to become even more difficult and complex. The main goal of DDoS defense mechanisms is to enable the victim to endure attack attempts without denying service to legitimate clients. This means that either, policies for resource consumption are enforced or abundant resources to the legitimate clients are ensured. Most of todays proposed countermeasures that address DDoS attack fall under two basic categories. Reactive defense involves detecting the presence of attack packets by using either anomaly-based detection or signature-based detection techniques; and response mechanisms, attempt to alleviate the damage caused by the attack by reducing the intensity of the attack, by blocking attack packets or localizing the source of the attack using trace-back methods [10].

However, considering a single defense mechanism will not comprehend the strength of protection since it is a continuous process. Furthermore, shortcomings attached to each existing countermeasures might leave loopholes within the defense strategy. This means that its wise to design a defense system with best suited combination of countermeasures as suggested in different approaches such as [52].

In this paper, we have proposed a new classification that incorporates a comprehensive survey of all the existing and upcoming defense techniques, although, we are not advocating any technique but attempting to depict all of them into a taxonomy. Existing published research papers on taxonomy of DDoS attacks and defense can be found in [1],[7],[15], and [40]. Section 3 is the outline of the taxonomy which is divided into: Preventive measures (Section 3.1) that prevents the eventual target of the DDoS attacks; Mitigation measures (Section 3.2) are the ones which lessen the probability of attacks; Detection strategy (Section 3.3) detects event of attacks; Reactive mechanisms (Section 3.4) are capable of responding to the on-going attacks; Post-attack analysis techniques (Section 3.5) are deployed post-attack for attack-forensics and appropriate recovery; Deployment Strategy (Section 3.6), defines the optimal placement of various defense

[1] In a flooding-based DDoS attack, large amount of malicious traffic is directed to the victim with intent to consume its resources and degrade its network performances, sometimes to a point where these networks cannot be used any further. These attacks can be: a) Single-source - when the victim is flooded by a single zombie. b) Multi-source - when several zombies are involved. However, classical metrics parameters, (such as throughput, number of packets or bytes) are used to detect such attacks.

[2] A reflector is a legitimate host that responds to requests (e.g. a web server), attacker spoofs victims IP address from the source field of the request, tricking it to direct its response to the victim.

components; Cooperative degree (Section 3.7) classifies the extent of coordination for effective defense strategy; and Evaluation (Section 3.8) of the defense infrastructure to assess its strength. The diagram of the taxonomy[3] is appended in the Appendix A.

3 The Proposed Taxonomy

3.1 Preventive Measures

Preventive measures can be applicable even in the absence of attack which aims for strengthening the eventual target (like operating system, protocols, applications, services, etc.), of an adversary. However, this implies fixing all the vulnerabilities, of all Internet hosts that can be misused for an attack. Nevertheless, it does not take away the benefit of deploying them to prevent the overall system to an extent that its not an effortless job for an attacker.

Zombie Prevention, preventing the attacker from constituting an army of zombie computers in the first place [32], to attain this goal, it is necessary to eliminate all the weaknesses that attackers might exploit to gain control of hosts connected to the public Internet and preventing the attack code from propagating. For example, this can be best achieved at the ISP level [48] where global data flow is visible.

Protocol Security, addresses the problem of misuse of protocol design, like TCP SYN Attack, IP Spoofing, malformed packet formation, authentication server attack, fragmented packet attack, etc [11].

System Security protects a machine against illegitimate accesses. Several layers of security certify the overall security. A) ***Data security*** keeps check on the integrity of the crucial data of the system for instance, configuration files of the system, DNS entries, cryptographic variables, and routing tables. B) ***Application security*** protects the applications running on a specific operating system from hooks, backdoors, software vulnerability, code alteration, trojan horse, rootkits, and virus. C) ***Securing resources*** like buffers, file descriptors, address space, disk space, CPU cycles, and bandwidth. D) ***Securing services*** from being deleted, turned off, hacked or tampered, hidden or misdirected, or changed location. E) ***Securing wireless network***, a wireless service might also face the same threat as a wired network would. All their inherent resource limitations are particularly susceptible to the consumption or destruction of these scarce resources. Some of the threats are: Jamming (deliberate interference with radio reception to deny the target's use of a communication channel), bandwidth exhaustion, collisions of the wireless signals, misleading routing paths, flooding messages, interfering the communication, etc. Recently, researchers are giving lot of attention to secure wireless network, where we can deploy extension of already existing techniques [34].

[3] In the proposed taxonomy, the same measures can belong to several sections according to our classification.

3.2 Mitigation Measures

Mitigation measures regulate the probability of an attack (like, introducing some form of strong authentication before any critical network resource is requested), and monitoring the system. This incorporates formulating protocols for resource management, DoS aware algorithms, traffic monitoring, and techniques that can be deployed in the event of attack.

Resource Accounting [41] ensures fair service to each entity (process, user, IP address, or a set of IP addresses) by enforcing privilege-based and behaviour-based access of resources. Given that each entity is tied-up with legitimacy-based access mechanisms to verify its identity.

Resource Multiplication [23] provides an abundance of resources to counter DDoS threats. For example, deploying a pool of servers with load balancer, and installs high bandwidth links between itself and upstream routers. Although, this is a cost amplifying mechanism, this approach has often proved sufficient to prevent the DDoS attack consequences.

Resource Pricing is a client-oriented defense scheme where it is charged for computational and monetary resources forcing them to regulate their traffic [29].

DoS Aware Algorithms are simple algorithms when implemented along with the operating system (OS) could help mitigate the effects of a DDoS attack like Bin Selection algorithm [14]. E.g., scanning TCP-connection-queue at regular interval and dropping all half-open connections could prevent TCP-SYN attack.

Traffic Flow Monitoring, NetFlow[4] provides valuable information about network users and applications, peak usage times, and traffic routing like Cisco NetFlow [6]. This can be used to monitor the flow of traffic. Not only should it be capable to identify the malicious traffic but also flash crowd[5]. Bro [35] is a publicly available Network Intrusion Detection System (NIDS) that passively monitors a network.

Traffic Volume Monitoring, a sudden growth in the traffic volume is an indication of DDoS attack. There are two algorithms for identifying the large flows: (a) Sample and hold, and (b) multistage filters [51]. Some methods exist for measuring the traffic, like MULTOPS (MUti-Level Tree for Online Packet Statistics) [15] which rely on the assumption that the packet rate of incoming and outgoing traffic is proportional. While, making it unsuitable for asymmetric routers, e.g. some real video stream network. Since MULTOPS utilizes a tree structure to detect ongoing bandwidth attacks it fails to detect the attack launched from distributed sources. Besides, IP spoofing also affects the capability of MULTOPS.

[4] Netflow is defined with unique seven attributes: Source and Destination IP address, Source and Destination port, Layer 3 protocol type, TOS byte, Input logical interface.

[5] Severe resource consumption may also occur as a result of perfectly legitimate activity, resulting in a flash crowd when numerous legitimate users access a popular service simultaneously.

Source IP Address Monitoring [17] is an effective way to distinguish the kind of traffic (flash flow or DDoS attack). Usually, the source IP addresses in DDoS attacks are unknown while IP addresses are known in other cases.

Monitoring Other Features, besides flow, volume and IP address monitoring, there are other characteristics of DDoS attacks that could be detected, such as content of the packet and IP header, ramp up behaviour for multisource attacks, and spectral content [49].

IP Hopping [42] Network Address Translation gateways are used when server changes its IP address without changing its physical location, clients use DNS to look up its IP address. All packets destined to the old IP address are then filtered-out. However, it does not block a persistent attacker which continuously looks up for new IP address using DNS.

Load Balancing, balancing the load to each server in multiple-server architecture can improve both normal performances as well as mitigate the effect of a DDoS attack [43].

TCP-Migrate [44] TCP Migrate options support the migration of an active TCP connection across IP addresses by first establishing a secure connection (using secret cryptographic cookie negotiated through an Elliptic Curve Diffie-Hellman exchange) and then sending a new SYN packet from the desired new IP address with the Migrate option enabled.

Mutable Services [45] is a framework that allows relocating a service -and informs pre-registered clients of the new location through a secure DNS service.

3.3 Detection Strategy

Detection strategies render the visibility of attacks. There are, basically two kinds of detection schemes (a) Pattern detection; and (b) Anomaly detection. Precisely, the motive of these strategies is to characterize legitimate traffic and malicious traffic. Thus keeping a perfect balance between the false positives and false negatives; and true positives and true negatives [26].

Pattern Detection looks for the presence patterns (aka Signature) of the known attacks by constantly monitoring each communication against stored signatures in a database. Apparently, they are capable of efficiently detecting known attacks with almost no false positives, whereas new attacks or even slight variations of old attacks might be unobserved.

Anomaly Detection relies on a prototype of system behaviour at normal conditions known as normal traffic dynamics or expected system performance, which is periodically compared with the current models to detect anomalies, allowing it to discover unknown attacks as well. The caveat is a trade-off between the precise detection, and tendency to misidentify abnormal behaviour as normal. A) ***Standard*** - They rely on protocol standard for example, TCP protocol specification describing a three-way handshake for TCP connection setup, subsequently, an attack detection mechanism can detect half-open TCP connections. Since it depends on the protocol itself it doesn't spawn any false positives. Nevertheless, attackers can perform sophisticated attacks that seem to be compliant to the standard and may pass undetected. B) ***Trained*** - They generate specifications of system behaviour and traffic in normal conditions to specify allowed threshold values for different parameters. They can catch a broad range of attacks but with disadvantages: (a) Threshold setting: Setting a low threshold leads to many false positives, while a high threshold reduces the sensitivity of the detection system. (b) Model update: Systems and communication patterns evolve with time, and models need to be updated to reflect this change. Usually, they perform automatic model update using statistics gathered at a time when no attack was detected. However, this approach makes the detection mechanism vulnerable to attacks with slow-increase-rate that might misguide prototype and delay or even avoid attack detection.

3.4 Reactive Mechanisms

Reactive mechanisms [2] are capable of responding in an appropriate way to an ongoing attack, by ceasing the impact of an attack or tracing back to identify the attacker. They require having certain degree of spontaneity to cope up with the vast incoming stream of malicious traffic.

Filtering [46] - An effective way to impede DDoS attacks is to drop the packets characterized as unwanted or malicious. Although, some DDoS attacks use packets that request legitimate services making them non-filterable. Nevertheless, they prove useful in defending against the spoofed IP packets. Unless the characterization is accurate, filtering mechanisms run the risk of accidentally denying service to legitimate traffic. In a worse case clever attackers might leverage them as denial-of- service tools. A) ***Ingress/ Egress Filtering*** - From the deployment perspective, filtering can be ingress and egress [12]. Ingress filtering drops spoofed incoming packets and is deployed on the external interface of a network. While Egress filtering drops all outgoing spoofed packets and is deployed internally. However, it cannot eliminate the packets with spoofed addresses valid in the local internal network, or attacks packets that do not use spoofed IP addresses. B) ***Time-window based packet filtering*** [47] exists before the regular queue management operation in a router. Based on a sliding time-window-size which is dynamically changed, it identifies and drops malicious and aggressively increasing attack flows. C) ***History based IP filtering***, [5] Edge routers save all IP addresses which have been proved to be legitimate in its previous connection history. When victim is suffering from a high level of congestion, packets from IP addresses that do not exist in the database are dropped. D) ***Hop-Count based filtering*** [13] supports the fact that although an attacker can forge any field in IP header but the number of hops (to reach the destination) of an IP packet can't be falsified. By monitoring Time to Live (TTL) attribute from the IP header it identifies likely spoofed packets. E) ***Statistical approaches*** [20] keeps the statistics of IP header attributes (such as IP address, TTL, protocol type etc.), to deem most likely attack packets which are later dropped. F) ***Route based distributed packet filtering*** [33] uses routing information to determine if a packet arriving at a router is valid with respect to its inscribed source/ destination addresses, given the ability to reach constraints imposed by routing and network topology. Statistical En-route Filtering

(SEF) technique can detect false alarms as it requires the routing information be validated by the generating node using multiple keyed MAC. When forwarded, nodes along the way, verifies them and drops those with invalid MACs at earliest points. G) ***Protocol based filtering*** [4] drops all malformed packets, e.g. TCP packets with zero data size, or unusually large ICMP packets. H) ***Packet filtering*** at routers eliminates clearly-defined attack signatures, such as obviously wrong source addresses. I) ***Adaptive packet filtering*** [21] provides differential QoS for attack and valid traffic. In this mechanism routers create time-based counter for each packet they forward. Legitimate packets will have higher values as they appear regularly while spoofed IP addresses will turn up only during DDoS attacks.

Rate Limiting - These methods enforces limiting suspicious packets as marked by the detection mechanism in the situations where they have many false positives or cannot accurately characterize the malicious traffic. Nevertheless, highly sophisticated attack-traffic might still slip out. ***Max-Min Fair Share*** sets up routers to access a server with logic of minimum and maximum allowed share. This way incoming traffic is to the level safe for the server to process. ***Level-K*** controls the traffic admission rates of the routers k hops away from the victim using a max-min fairness approach [16].

Reconfiguration [15] At the event of attack, changing the topology of the victim or the intermediate network to either add more resources to the victim or to isolate the attack machines, like reconfigurable overlay networks, resource replication services, attack isolation strategies, etc might help in ceasing the attack.

Active Traceback - IP traceback [9] allows tracing IP packets to their origins, bringing together the actual path and it may provide evidences of true source. Active traceback helps to locate the zombies and master who initiated the attack. It can be further divided into two categories: ***History-based traceback*** requires routers to store specific information about the packets passing through them and distribute it to its peers. ***Memory-less traceback*** discovers the route from the IP header of the attack packets [18]. ***Congestion Control (CC)*** Apparently, DDoS attacks congests a network i.e. disproportionate share of bandwidth to entities. CC regulates the behaviours of network flow at different levels of abstraction: Link, Flow and Aggregate [27].

Link Based CC, a router maintains a queue for each incoming link. Forwarding of packets is done by sampling the head packet from each queue on a round-robin basis.

Flow - Routers can be configured to throttle certain flows instead of a specific ingress link. This can be refined into RED, FRED, SRED, RED-PD [18]. ***Random Early Detection*** CC technique works for packet-switched networks. When queue size exceeds a preset threshold, gateway marks arriving packets with a certain probability to be verified later. ***Flow Random Early Drop*** provides state-based selective dropping based on per-active flow buffer counts. It prevents unfair share of aggressive flows from monopolizing buffer space and bandwidth.

Stabilized Random Early Drop technique discards packets having high hit (arriving packet is from the previous flow) ratios. ***RED with Preferential Dropping*** uses the packet drop history at the router drops packets from upcoming high bandwidth flows. However, low-rate TCP- targeted DDoS attacks may escape both RED and RED-PD.

Aggregate is a collection of packets (TCP SYN, ICMP, etc.) from one or more flows that have some property in common. Two Aggregate-Based CC (ACC) mechanisms commonly used are: Local ACC and Pushback [3]. ***Local ACC*** provides an entirely self-contained solution at a single router for detection of early signs of congestion and rate-limiting of high-bandwidth aggregates. ***Pushback*** extends local ACC with communication and coordination capabilities. It allows a router to request adjacent upstream routers to rate-limit the specified aggregates. Selective Pushback sends rate- limiting requests to routers sending traffic with higher than normal rates. The detection of these routers and the profiling of normal traffic are performed via an enhanced probabilistic packet marking scheme.

Agent Identification - A mechanism for reliable and accurate agent identification would be necessary for liability enforcement. This motivates deployment of DDoS defense far from the victim network.

Deflection - This section classifies the techniques deployed to draw away the attackers target, meanwhile gather all the vital information about the on-going attack. A) ***Honeypots*** [27] are machines that are not supposed to receive any legitimate traffic. Traffic destined to a honeypot is probably an ongoing attack that can be analyzed to reveal vulnerabilities targeted by attackers. Traditional honeypots are deployed at fixed, detectable locations which can be dodged by sophisticated attacks. Monitoring architecture deploys low-interaction honeypots as the frontend content filters and high-interaction honeypots to capture detailed attack traffic. Shadow real network resources, is a virtual Honeynet system that has a distributed presence and centralized operation, camouflaging real network resources, disguising the attacker to have conquered the network resources. B) ***Study the attack***, honeypots run special software which constantly collects data about the system behaviour for automated post-incident forensic analysis. Honeypots are allowed to be compromised and behave as a normal machine, silently spying valuable information about the activities. This also helps in retrieving some critical information like identifying the communication channel used for the attack, copy of the attack code, etc. C) ***Roaming honeypots*** can be deployed at service-level, where the locations of honeypots are spontaneously changing within a pool of back-end servers.

Alerts and Reports - Recent trend towards self-healing systems that automatically detects, diagnoses, and respond to failures has recognized the merits of generating alert and report precisely. Automated alerts/responses are generated with little or no user guidance. Adaptive, updates the responses on-the-fly on the basis of their success and failure in thwarting previously seen intrusions. Pre-emptive, triggers responses even before the attack completes. Cost-Sensitive, is similar to adaptive but assesses the risk and cost of responding or ignoring a warning, before generating the response.

Data Recording (Logging) - Network components (such as firewalls, packet sniffers, log-servers) records the incident details that might discover some crucial information about the attack, during forensics analysis. If attacker has done considerable financial damage, it also assists in Law enforcement.

3.5 Post Attack Analysis

Data recorded during a DDoS attack can be analyzed for some specific characteristics which can be used as a feedback for the defense system to enhance their efficiency and protection ability e.g. updating filtering parameters.

Forensics of Event and System Logs - Logs from the entire duration of attack can be used for forensic analysis. Custom-made sniffers and scripts help to trace the activity of malicious software and will also reveal type of attack, communication channels used, impact on the victim, affected applications, etc.

Passive Traceback works when the attack is on its completion, so it can only disclose the zombie army deployed for the attack. For this reason it is placed in this section. It can be performed manually or recursively, until the source is traced.

Traffic Pattern Analysis uses forensics logs. NetState [4] includes sniffer modules that passively monitor traffic across a network, and stores information (like, service name and versions) in a backend database that can be retrieved via GUI client.

Attacker Identification is identifying the master-mind behind the attack. Sophisticated attackers come up with mechanisms that outwit existing signature-based detection and analysis techniques. However, some communication link between the attackers and a compromised host will certainly lead to a trail.

Attack Reconstruction is process of presuming which entities carried the attack forward. This may not be visible when any single host is observed but when viewed globally. It identifies the hosts, compromised along the way, and highlights the crucial information about the attack propagation.

Updating the System - After the attack has been analysed it can be used as a feedback to the existing system, for instance, a new attack signature or a loophole.

3.6 Deployment Strategy

The deployment strategy must be globally attuned with Source Address Validity Enforcement Protocol (SAVE).

Victim-End Defense will prevent it from DDoS attacks and responds by alleviating the impact, e.g. resource accounting and protocol security mechanisms. ***Path Signature*** (PS) based system [31] uses a deterministic packet marking scheme, Path Identifier (Pi), that uniquely identifies the path took by each packet. ***Net Bouncer*** [24] maintains a legitimacy list, created after client has proved its legitimacy on the challenges presented to it. It makes use of various QoS techniques to assure fair sharing of resources that too expires after a certain interval. ***XenoService*** [23] is a distributed network of web hosts who dynamically adapt to flooding attacks, when under attack it replicates rapidly to absorb packet flood.

Source-End Defense prevents network customers from generating DDoS attacks. ***D- Ward*** [22] monitors each peer for signs of communication difficulties. Periodically, compares it with a prototype of normal traffic and imposes rate limit methods on the suspicious outgoing flow, when suspects DDoS attack. ***Source Router Preferential Dropping*** (SRPD) mechanism [8] is not a pure source-end DDoS defense system because it needs the victim-end to send a newly designed ICMP response message that contains queue occupancy rate. In high-rate flow, SRPD reacts by dropping packets with a probability calculated based on the average response time.

Distributed Defense offers scalability, often required by the system using multiple entities (routers, computers, firewalls) for protection. Including more defensive nodes enhances the defense strength and makes an attackers job substantially difficult as each entity plays a specific role [25]. The architecture of an ***Overlay-based*** approach, Secure Overlay Services (SOS) [28] proactively prevents DDoS attacks by forming a network overlay. At the overlay entrance, the clients are authenticated. At any instance only a small, random set of clients are allowed to reach the server while all other traffic is filtered out. A non-parametric ***Cumulative Sum*** (CUSUM) scheme [36] with low computational complexity helps in building accurate statistics to describe the pre-change and post-change in traffic distributions and monitors the short-term behaviour shifting from a long-term behaviour. Once the cumulative difference reaches certain threshold, an attack alert is raised. ***Defensive Cooperative Overlay Mesh*** (DefCOM) [22], deploys defense nodes distributed in the Internet core, forming a peer-to-peer overlay to securely exchange attack-related messages. It consists of three types of nodes: alert generators (collects detection statistics from each node and floods alert messages to rest of the nodes), classifiers (differentiate between legitimate and attack packets), and rate-limiters (controls attack traffic at source-end routers).

3.7 Cooperative Degree

Defensive measures can either act alone or in cooperation with other entities in the Internet. Attack detection proves to be accurate near the victim, while response and traffic classification is more successful near the source.

Autonomous defense mechanism defends only at the point where they are deployed (a host or a network), e.g. Antivirus. Defense at the ***intermediate*** network provides infrastructural service to a large number of Internet hosts e.g. Pushback and traceback techniques. They are yet to be widely deployed. ***Cooperative*** defense is autonomous at attack detection and response while rate-limit requests can be propagated to peers, e.g. pushback mechanism.

Distributed defense consists of two key stages: First, each defense node detects traffic anomalies according to its local defense policy using a variety of existing IDS tools which may have high false positives. Second, enhance the accuracy of the defense mechanism by using gossip-based communication mechanism [50] to share information among the defense nodes. The information gathered from individual defense node will have approximate global statistics of the attack behaviour helping to defend against attack traffic more efficiently by dropping the traffic with higher accuracy. ***Distance-based*** defense system coordinates with victim and source ends. Victim end detects an attack and a traceback component analyzes the attack traffic to find the addresses of remote routers forwarding attack traffic. An alert message will be sent to the source-end defense systems which are in charge of these routers. ***Global DDoS Defense Infrastructure*** (GDI) [39] employs a distributed architecture consisting of detection and traffic-filters installed in transit network routers and only few of them at the stub. Anomaly detection algorithm detects attack traffic based on traffic volume, and a traffic threshold algorithm identifies the network interfaces involved. ***Interdependent*** defense requires deployment at multiple networks or rely on other entities for attack detection, prevention, and response. E.g., Secure overlay services.

3.8 Evaluation

Evaluation of any defense system not only helps to quantize its strength but also helps in determining the optimal tradeoffs. ***Accuracy***, a defense system should be able to classify malicious traffic (false negatives and true positives) and legitimate traffic (false positives and true negatives) accurately, while keeping the damage minimum. Low false positives offer effectiveness of the system whereas low false-negatives present a measure of the system reliability. False positive rate measures the percentage of legitimate packets dropped by the rate limiting mechanism, and false negative rate measures the percentage of attack traffic pass the defense node. ***Effectiveness***, regardless of whether the attack is disruptive or degrading how successfully a defense mechanism can cease the attack and get the system up and running again. ***Transparency*** of the defense techniques is required before deployment. E.g., the deployment of pushback requires modification of existing core routers and likely purchase of new hardware. New components must be ***Compatible*** with the existing infrastructure. An estimation of the extent of the modifications required while installation at the client-side should be known beforehand. It should be reasonably cheap with excellent ***performance***, in terms of identifying DDoS attacks accurately, in the presence of attack or not. Before deployment a defense system must be analyzed if it is ***vulnerable*** to DDoS attacks. The defense system should be practically ***deployable*** having its deployment strategy thoroughly analyzed. E.g., deploying pushback mechanism on few core routers can affect many traffic flows. ***Scalability*** is crucial for any real-world countermeasure; it should be possible to extend a defense system when additional resources need to be protected.

4 Conclusion

In this paper, we have performed an extensive survey of the existing, all possible, parameters that might affect the defense strategy of a system; and we have proposed an enhanced autonomous taxonomy which represents the current state-of-the-art. Earlier publications considered the taxonomies with reactive and mitigation mechanisms, later some included deployment strategies and forensics as part of the strategy. Whilst it is important to understand that defense strategy is continuous process. Our taxonomy gives an insight of almost all the components that are necessary to complete a defense strategy, keeping in mind, its spirally growing architecture. It also emphasizes the required properties of such components, which comprise autonomy, self-configuration, dynamism and adaptation. Lets say, while building a strategy we ignore the post-attack analysis then we might not be able to construct new rules for the attack that was just experienced by the system. Our focus in this paper restricts to bring together all the aspects that affect the defense infrastructure while we exclude the discussion of advocating individual mechanism. However, this will serve the research community to visualize all the components might comprise an autonomous DDoS defense strategy.

References

1. Specht, S.M., Lee, R.B.: Distributed denial of service: taxonomies of attacks, tools and countermeasures. In: Proceedings of the 17th ICPADS, pp. 543–550 (2004)
2. You, Y., Zulkernine, M., Haque, A.: Detecting Flooding-based DDoS attacks. In: Proceedings of IEEE International Conference on Communications, pp. 1229–1234 (2007)
3. Loannidis, J., Bellovin, S.: Implementing pushback: router-based defense against DDoS attacks. In: Proceedings of the Network and Distributed System Security Symposium (2002)
4. Daniels, T.E., Spafford, E.H.: Network Traffic Tracking Systems: Folly in the Large. In: Proceedings of the 2000 Workshop on New Security Paradigms, pp. 119–124 (2000)
5. Peng, T., Ramamohanarao, K., Leckie, C.: Protection from distributed denial of service attacks using history-based IP filtering. Proceedings of the IEEE 1, 482–486 (2003)
6. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
7. Lau, F., Rubin, S.H., Smith, M.H., Trajkovic, L.: Distributed denial of service attacks. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2275–2280 (2000)
8. Fan, Y.: Defeating Denial of Service attacks with source router preferential dropping. Master thesis: Queens's University, Kingston Canada (2003)
9. Savage, S., Wetherall, D., Karlin, A.R., Anderson, T.: Practical network support for IP traceback. In: SIGCOMM, pp. 295–306 (2000)
10. Carl, G., Kesidis, G., Brooks, R.R., Rai, S.: Denial of-service attack-detection techniques. IEEE Internet Computing 10(1), 82–89 (2006)
11. Chang, R.K.: Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. IEEE Communications Magazine 40(10), 42–51 (2002)
12. Ferguson, P., Senie, D.: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. United States (2000), http://rfc.net/rfc2827.html

13. Jin, C., Wang, H., Shin, K.G.: Hop-count filtering: an effective defense against spoofed DDoS traffic. In: Proceedings of the 10th CCS 2003. ACM, New York (2003)
14. Sherr, M., et al.: Mitigating DoS attack through selective bin verification. In: Proceedings of IEEE ICNP Workshop, pp. 7–12 (2005)
15. Mirkovic, J., Reiher, P.: A Taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review 34(2), 39–53 (2004)
16. Yau, D.K.Y., Lui, J.C.S., Liang, F., Yam, Y.: Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router Throttles. IEEE/ACM (TON) 13(1), 29–42 (2005)
17. Peng, T., Christopher, L., Kotagiri, R.: Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring. In: IEEE Infocom 2004, Hong-Kong (2004)
18. Xiang, Y., Zhou, W., Chowdhury, M.: A survey of active and passive defense mechanisms against DDoS attacks. TR C04/02, Deakin University, Australia (2004)
19. Mls, J.: Effectiveness of rate-limiting in mitigating flooding DoS attacks. In: Proceedings of the Third IASTED International Conference, pp. 155–160 (2004)
20. Feinstein, L., Schnackenberg, D., Balupari, R., Kindred, D.: Statistical approaches to DDoS attack detection and response. In: Proceedings of the DARPA, vol. 1, pp. 303–314 (2003)
21. Dubendorfer, T., Bossardt, M., Plattner, B.: Adaptive distributed traffic control service for DDoS attack mitigation. In: Proceedings of 19th. IEEE, Los Alamitos (2005)
22. Mirkovic, J., Reiher, P.: D-WARD: a source-end defense against flooding denial-of-service attacks. IEEE Transactions on Dependable and Secure Computing 2(3), 216–232 (2005)
23. Yan, J., Early, S., Anderson, R.: The XenoService: a distributed defeat for distributed denial of service. In: Proceedings of ISW (2000)
24. Thomas, R., Mark, B., Johnson, T., Croall, J.: NetBouncer: client-legitimacy-based high-performance DDoS Filtering. In: Proceedings of the DARPA, vol. 1, pp. 14–25 (2003)
25. Mirkovic, J., Robinson, M., Reiher, P., Oikonomou, G.: A framework for collaborative DDoS defense. In: Proceedings of ACSAC (2006)
26. Carl, G., Kesidis, G., Brooks, R., Rai, S.: Denial-of-service attack detection techniques. IEEE Internet Computing 10(1), 82–89 (2006)
27. Champagne, D., Lee, R.B.: Scope of DDoS countermeasures: taxonomy of proposed solutions and design goals for real-world deployment. Princeton Univ. Tech. Report CE-L2005-007 (2005)
28. Keromytis, A.D., et al.: SOS: secure overlay services. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 61–72 (2002)
29. Mankins, D., et al.: Mitigating distributed denial of service attacks with dynamic resource pricing. In: Proceedings of the Computer Security Applications Conference, pp. 411–421 (2001)
30. Hu, Y.H., et al.: Packet filtering for congestion control under DoS attacks. In: Proceedings of the 2nd IEEE Int. Information Assurance Workshop, pp. 3–18 (2004)
31. Yaar, A., et al.: Pi: a path identification mechanism to defend against DDoS attacks. In: Proceedings of the 2003 Symposium on Security and Privacy, pp. 93–107 (2003)
32. Dalton, M., et al.: Real-World Buffer Overflow Protection for User-space and Kernel-space. In: Proceedings of the 17th conference on Security symposium, pp. 395–410 (2008)

33. Park, K., Lee, H.: On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets. In: Proceedings of ACM SIGCOMM 2001 (2001)
34. Lough, D.L.: A Taxonomy of: Computer Attacks with Applications to Wireless Networks. PhD thesis: Virginia Tech, Computer Engineering Department (2001)
35. Paxson, V.: Bro: A system for detecting network intruders in real-time. In: Proceedings of the 7th Annual USENIX Security Symposium, San Antonio, Texas (1998)
36. Pollak, M.: Optimal detection of a change in distribution. Ann. Statist. 13, 206–227 (1986)
37. Cheng, C.M., Kung, H.T., Tan, K.S.: Use of Spectral Analysis in Defense Against DoS Attacks. In: Proceedings of GLOBECOM 2002, vol. 3, pp. 2143–2148. IEEE, Los Alamitos (2002)
38. Sourcefire. Snort: The Open Source Network Intrusion Detection System, http://www.snort.org/
39. Wan, K.K.K., Chang, R.K.C.: Engineering of a global defense infrastructure for DDoS attacks. In: Proceedings of the IEEE International Conference on Networks, pp. 419–427 (2002)
40. Usman, T.: A Comprehensive Categorization of DDoS Attack and DDoS Defense Techniques. In: Li, X., Zaïane, O.R., Li, Z.-h. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 1025–1036. Springer, Heidelberg (2006)
41. Kargl, F., Maier, J., Weber, M.: Protecting web servers from Distributed Denial of Service attacks. In: Proceedings of the 10th International Conference on WWW, Hong Kong, pp. 514–524 (2001)
42. Jones, J.: Distributed Denial of Service Attacks: Defenses. A Special Publication: Technical report, Global Integrity (2000)
43. Lan, Z., Taylor, V.E., Bryan, G.: Dynamic Load Balancing of SAMR Applications on Distributed Systems. In: Supercomputing, ACM/IEEE 2001 Conference Publication (2001)
44. Snoeren, A.C., Balakrishnan, H., Kaashoek, M.F.: The Migrate Approach to Internet Mobility. In: Proceedings of the Oxygen Student Workshop (2001)
45. Dewan, P., Dasgupta, P., Karamcheti, V.: Defending against Denial of Service attacks using Secure Name resolution. In: Proceedings of SAM 2003 (2003)
46. Stephan, B.: Optimal filtering for denial of service mitigation. In: Proceedings of the 41st IEEE Conference on Decision and Control, vol. 2, pp. 1428–1433 (2002)
47. Hu, Y.H., Choi, H., Choi, H.A.: Packet Filtering to Defend Flooding-Based DDoS Attacks. In: Advances in Wired and Wireless Communication, IEEE/Sarnoff Symposium, pp. 39–42 (2004)
48. Gupta, B.B., Misra, M., Joshi, R.C.: An ISP Level Solution to Combat DDoS Attacks using Combined Statistical Based Approach, pp. 102–110. JIAS: Dynamic Publishers Inc., USA (2008)
49. Hussain, A., Heidemann, J., Papadopoulos, C.: A Framework for Classifying Denial of Service Attacks. In: Proceedings of the ACM SIGCOMM Conference, Karlsruhe, Germany (2003)
50. Badishi, G., Keidar, I., Sasson, A.: Exposing and eliminating vulnerabilities to denial of service attacks in secure gossip-based multicast. In: The International Conference on DSN, pp. 223–232 (2004)
51. Estan, C., Varghese, G.: New directions in traffic measurement and accounting. In: Proceedings of the 2001 ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, pp. 75–80 (2001)
52. Debar, H., Thomas, Y., Cuppens, F., Cuppens-Boulahia, N.: Enabling automated threat response through the use of a dynamic security policy. Journal in Computer Virology 3(3), 195–210 (2007)

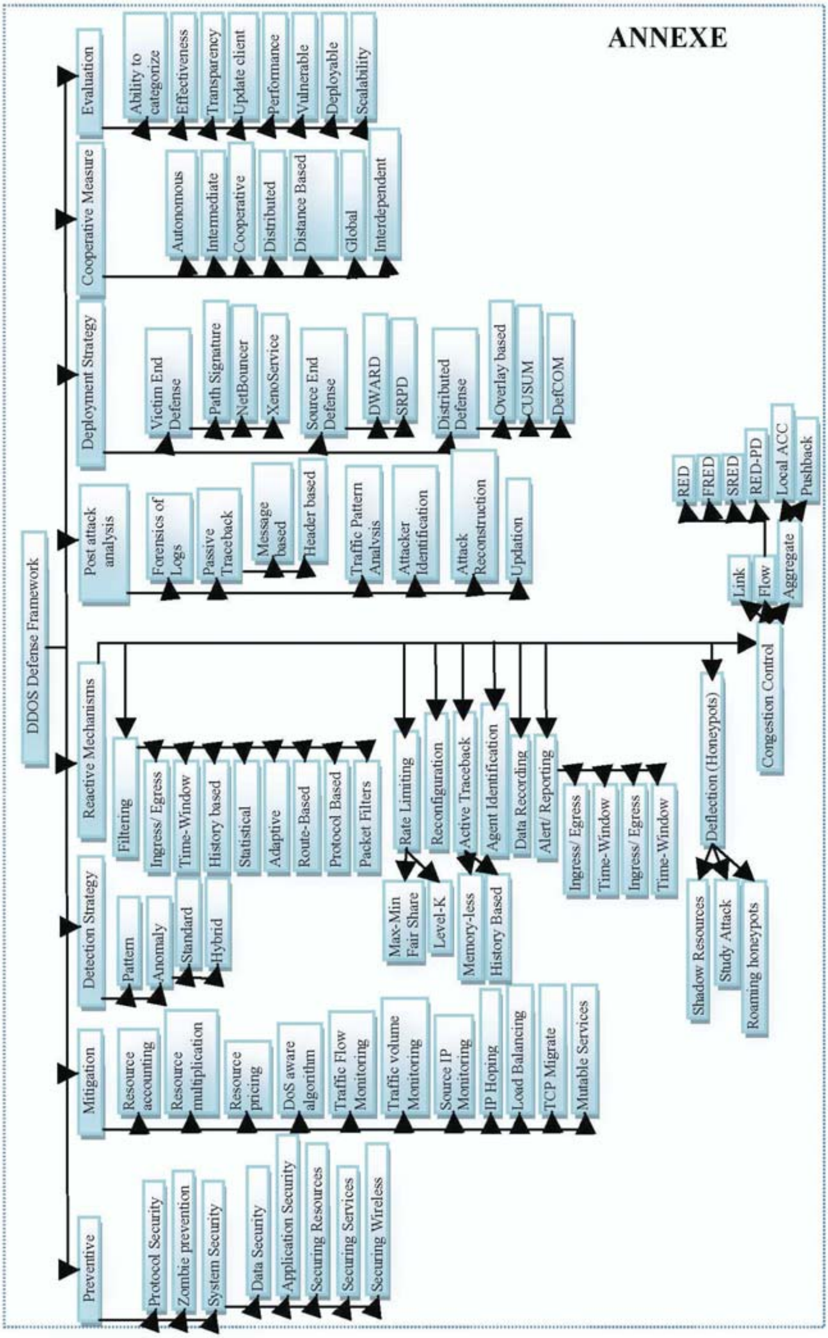


Fig. 1. DDoS Defense Mechanisms: A New Taxonomy

RDyMASS: Reliable and Dynamic Enforcement of Security Policies for Mobile Agent Systems

Houssem Aloulou[1], Monia Loulou[1,2], Slim Kallel[1], and Ahmed Hadj Kacem[1]

[1] Laboratory ReDCAD, B.P. W 3038 Sfax, Tunisia
{houssem.aloulou,slim.kallel}@redcad.org,
ahmed.hadjkacem@fsegs.rnu.tn
[2] Laboratory LaBRI, UMR CNRS 5800 Bordeaux, France
loulou@labri.fr

Abstract. Defining security policies is a crucial stage for an efficient implementation of security within mobile agent systems.To enforce these policies in a reliable way, it is necessary to make use of formal techniques which offer enough flexibility and expressiveness, and which provide a rigorous reasoning about the security of mobile agent systems security.

The migration of the agent between several systems can lead to inconsistencies between its policy and the policy of the system. These incoherencies may require a dynamic reconfiguration of the security policies of the system and the agent. This reconfiguration cannot be efficient without the use of mechanisms that improve the modularity of the security code.

In this paper, we aim to dynamically enforce security policies in mobile agent systems in modular and reliable way. For this purpose, we combine formal methods and aspect oriented programming. We propose a three-step approach to enforce security policies in the form of aspects that will be generated through a reliable specification of security policies.

Keywords: Security policy, Mobile agent systems, Dynamic enforcement, Aspect-oriented Programming.

1 Introduction

Security problems impede the expansion of the mobile agent technology. When an agent moves, it is vital to ensure that it will be correctly executed into the new visited system. Similarly, it is crucial to reassure the agent system that there won't be any risk when receiving a new agent. In order to master the complexity of the mobility and the security, it is necessary to use formal techniques, which offer enough flexibility and expressiveness, and provide a rigorous reasoning about the security of mobile agent systems.

Traditional approaches for the implementation of a security mechanism consist in going back to the application code and, manually, integrating the monitoring code in it. The adoption of such an approach constitutes a difficult task for the developer mainly in a dynamic context because it requires continuous changes

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 237–252, 2010.

in the definition of security policies. Moreover, this manual integration could not guarantee conformity with the formal policy specification. In other words, the code which implements the security policy could contain inconsistencies which do not exist in the specification level. The Aspect Oriented Programming (AOP) [1] allows to resolve all these limits. AOP suggests to separate the code of crosscutting concerns from the functional code of a software application, and implements them in separated modules that would be, afterwards, woven to constitute a complete application. In fact, the AOP extends traditional programming paradigms by providing complementary mechanisms in order to increase the modularity of applications.

The use of the AOP paradigm for implementing security mechanisms [2] allows an easy maintenance and best reuse of security aspects without worrying about its environment and regardless of the field of the application. Indeed, the code quality is improved thanks to the simplicity and the modularity of AOP. Moreover, dynamic AOP allows adaptability of security policies at runtime without having to stop the running application.

To enforce security policies, we should deploy a security mechanism. Mainly, there exists three classes of security mechanisms [3] such as : static analysis [4], execution monitoring [5], and program rewriting [3]. Each of them has some advantages according to the kind of application in which we would enforce security properties. To properly approach the problem of security in mobile agent applications, it's necessary to deploy a mechanism, which maintains a high-level availability of such applications, and which respects the dynamic aspect of their security requirements. For this purpose, we dismiss the static analysis and the program rewriting mechanisms because they operate on the program before its execution. Contrarily, the execution monitoring mechanism controls the behavior of a program at runtime and intervenes when an imminent violation of the policy is detected. This intervention consist in interrupting the execution of the method/action that violates a policy. The powerful form of intervention consists in inserting actions or omitting risky actions on behalf of controlled programs. This form of intervention corresponds to the rewriting-based execution monitoring mechanism. Thus, we adopt the rewriting-based execution monitoring mechanism to define a framework for the enforcement of security policies.

In this paper, we propose, first, to integrate AOP paradigm into an operational framework for dynamic security policy enforcement, which operates according to the rewriting-based execution monitoring mechanism. Second, we define a generative aspect-based approach that, automatically, generates the corresponding security aspects to the reliable specification of security policies. Thus, we will adopt a formal security framework for mobile agent systems which brings more details to the concepts emerging from combining security with agent mobility, and which treats the different concerns of security.

The remaining parts of this paper are organized as follows. Section 2 provides some background information on the Z notation, Aspect-Oriented Programming, and *JBoss AOP* tool. Section 3 gives a short overview of the adopted formal security framework. In Section 4, we give in details the three-steps of our approach

RDyMASS to have a reliable and dynamic enforcement of security policies for mobile agents system. In Section 5, we implement our approach on the practical context of the dynamic aspect weaver *JBoss AOP*. Section 6 presents an experimentation of our approach in order to secure an electronic transaction within a mobile agent-based application developed with Aglets. We discuss some related works in Section 7. Finally, we conclude this paper by summarizing our ideas.

2 Background

2.1 Z Notation

The Z notation, as presented in [6], is a model oriented language based on the set theory and first-order predicate logic. The Z language is distinguished by a schema language that provides a structured description of the system states and potential operations under which system state can change. Thus, a Z specification describes the static and dynamic aspect of the system. In order to verify a number of properties, a Z specification can be the object of assisted or automated formal proofs. To edit and prove Z specifications, we use Z/EVES tool [7], which ensures syntax and type checking and general theorem proving.

2.2 Aspect Oriented Programming

The principal characteristic of Aspect Oriented Programming (AOP) [1] is the separation of concerns into two categories: (i) functional concerns that present business code and (ii) technical concerns (aspects) that correspond to non-functional requirements. The separation of different types of concerns improve the modularity of applications. An aspect is formed by a *pointcut* and an *advice code*. A *pointcut* may involve one or more aspects. A pointcut is composed of one or many *joinpoints*. A *joinpoint* can capture specific events where an aspect can be weaved. An *advice code* is a mechanism, similar to a method, used to codify the code to execute in all joinpoints of the corresponding pointcut. The advice code can be executed *before*, *after* or *around* a joinpoint.

To get the application that integrate functional and technical concerns an *aspect weaver* is used. Two types of weaver exist: *static weaver* ensures weaving before starting the execution of the application, and *dynamic weaver* guarantees the weaving at runtime (during the execution of the application).

2.3 JBoss AOP

JBoss AOP [8] is a dynamic weaver which ensures the *weaving* and *unweaving* of aspects at runtime. Aspects are written in pure Java and use an API for JBoss AOP. To use JBoss AOP dynamically, two concepts are used: (i) the *Hot Deployment*, which is provided in a library of JBoss AOP in order to be able to weave and unweave aspects at runtime. (ii) the *HotSwap*, which configures the execution arguments of the application.

A JBoss AOP aspect consist of three files:

- The first represents the *advice*, named also *Interceptor*. It is encapsulated in a Java class that implements the interface *org.jboss.aop.Interceptor*. An interceptor contains the method *invoke* which encapsulate the advice code.
- The second file represents the *pointcut* defined as an *XML file* named *jboss-aop.xml*. It is required to prepare the *HotDeployment* of aspects at runtime. The tag used is "*prepare*". The use of the XML allows a high quality and a short time of weaving.
- The third contains the code which ensures aspects *weaving* or *unweaving* at *runtime* using respectively the method *addbinding* or *removebinding*.

3 Formal Security Framework

We present in this section a formal security framework for mobile agent systems [9] that support, through a specification framework, the expression of numerous security policy types in order to control the behavior of system entities and to protect them. In order to avoid any anomalies able to reduce the policy performance, its verification framework checks the consistency of the proposed specifications as well as the consistency intra-policy. All the proposed concepts, in this security framework, have been specified rigorously using Z notation [6] and checked using the Z/EVES toolkit [7].

3.1 Security Specification Framework

In a mobile agent system, execution environments (AgS) and mobile agent (MAg) should have well defined security policies in order to screen the incoming agents and/or adversary AgS respectively adversary MAg and hosting AgS. Thus, a secure entity *SEntity* can be either a MAg or an AgS.

$$SEntity ::= MAg \langle\langle MobileAgent \rangle\rangle \mid AgS \langle\langle AgentSystem \rangle\rangle$$

To express the various kinds of security policies, three basic constructs for authorization, prohibition, obligation are defined. Formally, this variety is specified as a free type:

$$SConstruct ::= Auth \mid Prohb \mid Oblig$$

Both mobile agents and agent systems aim to protect their secure objects denoted by (*SObject*). A secure object may be either data, or service or computing resource : $SObject ::= D\langle\langle Data \rangle\rangle \mid Sr\langle\langle Service \rangle\rangle \mid Rs\langle\langle CResource \rangle\rangle$.

Formally, a security rule is defined with the *SRule* schema below. The declaration part of this schema specifies the type of the security rule (*Type*), the secure entity concerned with the security rule (*Interested*), the subject entity (*RSubject*) on which we apply it, the target object (*Target*), the applicability context (*Context*) which designates the constraints that limit the applicability of the rule, and a non empty set of actions (*Actions*) to be enforced by the rule to reach the desired behavior.

$$\begin{array}{ll}
\multicolumn{2}{l}{\textit{SRule}} \\
\hline
\textit{Name} : \textit{Propriety} & \\
\textit{Type} : \textit{SConstruct} & \\
\textit{Interested} : \textit{SEntity} & \\
\textit{RSubject} : \mathbb{F}_1\, \textit{SEntity} & \\
\textit{Target} : \mathbb{F}\, \textit{SObject} & \\
\textit{Context} : \textit{Condition} & \\
\textit{Actions} : \mathbb{F}_1\, \textit{Action} & \\
\hline
\forall\, r : \textit{CResource} \mid \textit{Target} = \{\textit{Rsr}\} \bullet (\textit{Type} = \textit{Auth} \lor \textit{Type} = \textit{Prohb}) & \mathsf{[C1]} \\
\quad \land\, (\exists\, s_1 : \textit{AgentSystem} \bullet (\textit{Interested} = \textit{AgSs}_1 \land r \in s_1.\textit{Reserved\_res})) & \\
\qquad \land\, \neg\, (\exists\, s_2 : \textit{AgentSystem} \bullet \textit{AgSs}_2 \in \textit{RSubject}) & \\
\forall\, sc : \textit{Service} \mid \textit{Target} = \{\textit{Srsc}\} \bullet (\textit{Type} = \textit{Auth} \lor \textit{Type} = \textit{Prohb}) & \mathsf{[C2]} \\
\quad \land\, (\exists\, s_1 : \textit{AgentSystem} \bullet (\textit{Interested} = \textit{AgSs}_1 \land sc \in s_1.\textit{Services})) & \\
\qquad \land\, \neg\, (\exists\, s_2 : \textit{AgentSystem} \bullet \textit{AgSs}_2 \in \textit{RSubject}) & \\
\textit{Type} = \textit{Oblig} \Rightarrow \{\textit{Context}\} \neq \varnothing & \mathsf{[C3]} \\
\hline
\end{array}$$

The specification of a security rule must satisfy three constraints, given in the predicate part. For example [C1] states that when a target of a given rule is a computing resource, then the *Interested* entity in the rule must be an *AgS* and, indeed, the *RSubject* of the rule must be a *MAg*. Moreover, we check with [C1] that the AgS denoted with *Interested* can only control the access to its own resources. A complete description of the predicate part is presented in [9].

Formally, a security policy is specified with the following schema:

$$\begin{array}{l}
\textit{SPolicy} \\
\hline
\textit{Subject} : \textit{SEntity} \\
\textit{Rules} : \mathbb{F}\, \textit{SRule} \\
\hline
\forall\, r : \textit{SRule} \mid r \in \textit{Rules} \bullet r.\textit{Interested} = \textit{Subject} \\
\forall\, a, b : \textit{SRule} \mid a \in \textit{Rules} \land b \in \textit{Rules} \land a \neq b \\
\quad \bullet\, a.\textit{Name} \neq b.\textit{Name} \\
\hline
\end{array}$$

In the predicate part, we check that a policy *SPolicy* regroups the security rules which have the subject defined in the declaration part. Moreover, we check that different rules have different names.

3.2 Security Verification Framework

Writing proofs is an essential part in order to show the consistency of the specification framework and consequently improves the quality of the desired software [6]. In our context, the formal proofs can be considered at two different levels: proving the consistency of the specifications of the security policies and proving the consistency between rules of a security policy. We only present in this section how to formally prove the intra-policy consistency.

Conflicting and redundant security rules may reduce the performance of the policy and even make it inefficient. In fact, it is important to associate to the specification of security policies, a verification framework which check the two main cases of policy inconsistencies : the modality conflicts and the redundancy of rules.

Regarding the adopted specification of security policies, we distinguish three different modalities which are authorization, prohibition and obligation. Two types of modality conflicts may occur :

- an authorized action is forbidden by a prohibition rule,
- an obligation rule may require to perform an action which is forbidden by a prohibition rule.

For modeling the relationships which may exist between two or several rules, three relations has been defined: an unary relation named *Consistent* and two binary relations named *Contradictory* and *Redundant*.

In order to prove the consistency of a given policy, we should check that there is no contradiction between the policy rules and there is no redundant rules. On that basis, a rigorous definition of policy consistency given by a rewriting-rule *Def_Consistent* is defined. It appeals the definition of *Redundant* and *Contradictory* relations. A complete description of the specification of *Contradictory* and *Redundant* relations is presented in [9]. To prove the consistency of a given policy, require to defining a theorem which refers to the specification of the relation *Consistent*. Let's assume a security policy *Test_Policy*. To prove the consistency of *Test_Policy*, we add the following conjecture asserting:

$$\begin{array}{|l}
Consistent\_ \ : \mathbb{P}\, SPolicy \\
\hline
\forall\, p : SPolicy \bullet Consistent\ p \\
\Leftrightarrow (\forall\, a, b : SRule \mid a \in p.Rules \\
\quad \wedge\ b \in p.Rules \wedge a \neq b \\
\bullet \neg\ a\ Redundant\ b \wedge\ \neg\ a\ Contradictory\ b)
\end{array}$$

theorem $verif\_consistency$
$Consistent\ \ Test\_Policy$

The theorem's goal predicate is *Consistent Test_Policy*. When, we obtain the predicate true, after running a list of proof scripts, we prove that the conjecture is a theorem, and *Test_Policy*, indeed, is a consistent policy. A detailed example with regard to the proof of an intra-policy consistency is presented in [9].

4 RDyMASS Approach

We propose an aspect-based approach to dynamically enforce security policies in mobile agent systems. Our approach is based on [10] which propose to automatically generate aspect code from the formal specification of non functional safety properties, e.g. access control policies.

Our approach, baptized RDyMASS (*Reliable and Dynamic Mobile Agent System's Security*), enforces security policies according to the rewriting-based execution monitoring mechanism, which maintains high-level availability and respects the dynamic nature of security requirements. This mechanism respects the foundations of the dynamic AOP, which allow to weave or unweave aspects at runtime. To take advantage of the dynamic AOP, mainly in terms of modularity, we will adopt it in RDyMASS.

Our approach consists of a three-step process as schematically shown in Figure 1.

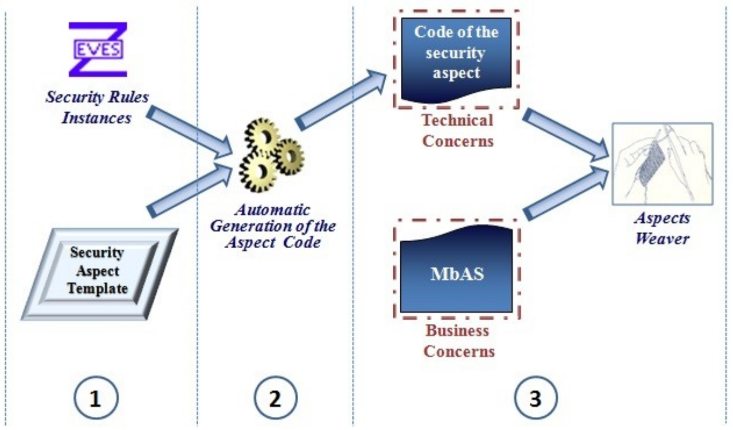


Fig. 1. An overview of RDyMASS approach

4.1 Aspect Template Step

To progressively reduce the gap between the security policy specification level and its implementation level, we begin by defining a *Template* for security aspects at two abstraction levels:

- The first level corresponds to the *generic aspect template*. It presents, at a high abstraction level, the skeleton of the security *aspect* regardless of the syntax imposed by the aspect weaver. This template will be defined according to the specification of security rule, presented in section 3.
- The second level corresponds to the *specific aspect template*. It is based on the generic aspect template and the terminology adopted by the chosen aspect weaver. Thus, we obtain a security aspect representation, which is close to the code.

The enforcement of a security policy is made possible by enforcing all its security rules instances. An aspect may encapsulate one or more instances of security rules at once. In order to simplify the structure of an aspect, to ensure a better adaptability, and to act separately on aspects, we choose to define one aspect for each security rule instance.

As presented in Listing 1, the generic aspect template consists of two main parts: a pointcut and an advice code.

The pointcut (lines 1–2) of the generated aspect intercepts the execution of mobile agents of the functional layer that corresponds to the attribute *actions* in a security rule instance. The pointcut contains the code of the crosscut. It includes the *joinpoint*, which is composed of one or more header methods corresponding to the attribute *Actions* of the class designating the *RSubject*.

The advice code (lines 5–21) checks the constraint specified in a security rule instance. If the check is successful, the aspect executes the action of the mobile agent, and after that updates the system state. Otherwise, the aspect prohibits the execution of the action. We use an *around* advice in order to allow or prohibit the execution of the agent action.

We decompose the advice into three subparts. The first part **[P1]** corresponds to the *before* part of the advice (lines 4–8). It is used to check constraints defined according to the *Context* attribute.

```
1  pointcut  pointcut_name = execution(public * RSubject's class ->
2  the corresponding method of the action attribute of the RSubject's  class)
3
4  [P1] ------------------------------------------------
5      //Checking the constraints presented in the Context signature
6      if(verifConstraint-1) {...}
7      ...
8      if(verifConstraint-n) {...}
9
10 [P2] ------------------------------------------------
11     if((All constraints are verified && Type of SRule is "Prohb")
12           || (There is one constraint not verified && Type of SRule is "Auth"))
13              // prohibit the execution of the corresponding method
14              // launch of an exception
15
16 [P3] ------------------------------------------------
17     else if((There is one constraint not verified && Type of SRule is "Prohb")
18          || (All constraints are verified && Type of SRule is "Auth"))
19              // implementation of the method that triggered the safety rule
20                   proceed(parameter)
21              // update of the system
```

Listing 1. Generic Aspect Template

Two cases may emerge. In first case the mobile agent is not authorized to perform the requested action. So, there will be a jump to [P2] (lines 10–14). This jump is realized when all constraints (checked in [P1]) are verified and the type of the security rule is *Prohibition* (Prohb), or when there exists at least one constraint that is not verified and the type of the security rule is *Authorization* (Auth). In this case, the request of the agent will not be granted and an exception will be launched. In second case the mobile agent is authorized to perform the requested action. So, there will be a jump to [P3] (lines 16–21). This jump is realized when all constraints (checked in [P1]) are verified and the type of the security rule is *Authorization* (Auth), or when there exists at least one constraint that is not verified and the type of the security rule is *Prohibition* (Prohb). In this case, the request of the agent will be granted by executing the method that triggered the security rule. Afterwards, an eventual update of the global system state will be done.

4.2 Aspect Generation Step

The second step of our approach RDyMASS is the automatic generation of the security aspect code. This generation will be done according to the syntax of the adopted aspect weaver. Contrary to the generic aspect template, that defines the structure of the security aspect regardless of the programming language, a specific aspect template refines this structure according to the adopted programming language (i.e. aspect weaver). Indeed, the specific aspect template will be used by the programmer to implement the corresponding generator code. This generator take the specifications of security rules instances as input elements and applies the imposed structure on the specific aspect template in order to generate the corresponding security aspects. These specifications will be imported from Z/EVES after proving their consistency.

4.3 Aspect Weaving Step

Once the aspect code is generated, we should weave this aspect into the functional code which represents a not secured mobile agent based application. This functional code should be implemented in concordance with the evoked concepts in the specification framework. There exist two types of weavers: a static weaver does the weaving action while compiling the application. On the contrary, dynamic weaver supports the weaving and unweaving of aspects at runtime. Therefore, dynamic weavers are more suitable for systems which require a high-level of availability and dynamic changes of their execution constraints such as the security requirements. Thus, a dynamic weaver should be used in our approach.

In this section, we have presented our approach RDyMASS, which ignores any choice of implementation. In the following sections, we will exhibit the integration of RDyMASS in a practical context related to a specific aspect weaver and an environment for the development and the deployment of mobile agent systems.

5 Implementation of RDyMASS

RDyMASS can be instantiated on several practical contexts that depend on the adopted dynamic aspect weaver. In this work, we implement our approach using JBoss AOP. The latter is free, in continual growth, and very well documented. It belongs to the family of JBoss server. There exist a standalone version which can be used independently of the server. JBoss AOP is also provided as an Eclipse plug-in that simplify the task of the programmer by a graphical user interface.

5.1 Specific Aspect Template for JBoss AOP

Once we set the aspect weaver, the following step consist in elaborating its corresponding specific aspect template.

According to the structure of a JBoss AOP aspect, already presented, the specific aspect template it similarly composed of three parts:

- *The pointcut:* it contains the same fields as the generic aspect template. The exception is the use of the keyword *prepare* in place of the *pointcut*. Thereby, specific pointcut for JBoss AOP is schematized by the Listing 2:

```
<aop>
   <prepare expr=”execution (public * RSubject’s class −>
   the corresponding method to the Actions attribute of the RSubject’s class)>
</aop>
```

Listing 2. Specific Aspect Template for JBoss AOP: Pointcut Part

- *The Interceptor:* It’s illustrated in the Listing 3. To present the progress of the interceptor’s execution, we decompose this template into three parts annotated [P1’], [P2’], and [P3’] which instantiate, according to JBoss AOP syntax, respectively [P1],[P3], and [P2] parts of the generic aspect template.

```
 import org.jboss.aop.advice.Interceptor;
 import org.jboss.aop.joinpoint.Invocation;

 public class Interceptor_Name implements Interceptor {
    public Object invoke(Invocation invocation) throws Throwable {

[P1’] −−−−−−−−−− Before part of the Interceptor −−−−−−−−−−−
    /*  Verification of constraints corresponding to the context attribute
    {Use of a Boolean constraint trait that takes the value ”true”
    if the constraints are verified and false otherwise} */

[P2’]−−−−−−−−−− Around part of the Interceptor −−−−−−−−−−−
    if (trait == true){
        System.out.println(”You are authorized to accomplish the request”);

        /* Invocation of the method corresponding to the action */
        Interceptor [] inter = new Interceptor[1];
        Class dynaclass = Thread.currentThread().getContextClassLoader()
                                                .loadClass(Resource_Class);
        Interceptor NewInterc = (Interceptor)dynaclass.newInstance();
        inter [0] = NewInterc;

        Object rsp = invocation.invokeNext();    // proceed in JBoss AOP

[P3’]−−−−−−−−−− After part of the Interceptor −−−−−−−−−−−
        /* Updating of the overall system state */
        return null;
    }
    else{
        System.out.println(”You aren’t authorized to accomplish the request”);
        return null;
            }
    }
}
```

Listing 3. Specific Aspect Template for JBOSS AOP: Interceptor code part

- *The binding code:* allows connection between a *prepare* tag and the interceptor. Listing 4 presents the structure of the binding code. The first line corresponds to the instantiation of an Advice Binding. It specifies the name of the method that triggers the security aspect execution. The fourth line presents the concerned interceptor. The latter will either be woven via addBinding (as presented in ligne 5) or be unwoven via removeBinding (as presented in ligne 6).

```
 1     /∗ instantiation of an Advice Binding ∗/
 2   AdviceBinding binding_Name = new AdviceBinding(”execution(public
 3   ∗com.ibm.awb.launcher.Main −> Method_Name)”, null);
 4   binding_Name.setName(”Name”);
 5    /∗ addition of the interceptor ∗/
 6    binding_Name.addInterceptor(Package.Interceptor_Name.class);
 7    /∗ add binding with the interceptor ∗/
 8   AspectManager.instance().addBinding(binding_Name);
 9    /∗ remove binding from the interceptor ∗/
10   AspectManager.instance().removeBinding(binding_Name);
```

Listing 4. Specific Aspect Template for JBoss AOP: Binding code Part

5.2 Automatic Aspect Generator

The second step is the generation of the JBoss AOP aspect code. It should be noted that the global application, which integrates the business concerns with technical concerns, will not take account of the updates (add/remove) of the security policy only after its redeployment. This is because the use of the *Java Virtual Machine* and the *dynamic weaving* [11].

The code generator is based on the specific aspect template of JBoss AOP to generate from an instance of security rule an equivalent aspect. The fields presented in bold in Listings 2, 3, and 4 will be automatically filled by the generator using the data extracted from the instance of security rule. At present, our automatic generator support only security rules of Auth/Prohb type and with *temporal constraint*.

Adding a new security aspect is achieved by: (i) the addition of a new Java file that will contain the interceptor, (ii) the addition in the Java file responsible of the binding of the code which provides *the dynamic adaptability* of security aspects, and (iii) the insertion of the tag *prepare* in the XML file. All these files will be added automatically, through the generator. Deleting an aspect is also, automatically, accomplished by deleting the file of the interceptor, the fragment of code responsible for the binding, and the tag ”prepare” in the XML file.

6 Experimentation

To show the efficiency of our approach RDyMASS, we present in the following an experimentation which consist in enforcing rules for securing electronic transactions within a mobile agent-based application. For this purpose, we adopt a deployment platform for mobile agent systems to implement the business concerns. Aglets is open source, free, light, and developed in Java. However, the level of Aglets security remains a major obstacle for its expansion. Therefore, we will try through our approach to ensure security in reliable way.

6.1 Aglets

Aglets [12] is a platform for developping mobile agent systems. The agents are presented as a set of Java objects that can move from one host to another on the network using the protocol *ATP* (Agent Transfer Protocol) to satisfy the requests for their owners. The Aglets Software Development Kit (ASDK) is provided with a graphical user interface named *Tahiti*, which facilitates the management of mobile agents.

The life cycle of an Aglets agent begins with its creation. During its lifetime, an Aglets agent may be transferred from one host to another, it can be also cloned, removed, enabled or disabled for a period of time.

6.2 Case Study: E-Commerce Secure Transaction

Nowadays, the development of the mobile agent-based applications is in continual evolution and metamorphosis. The exploitation of mobile agents in e-commerce applications is among the best alternatives for the problem of dispersal of sites on the network. First, we describe the architecture of an electronic-commerce mobile agent-based application. Second, we detail an example of security policy enforcement according to RDyMASS approach.

The case study consists of two major modules: *selling sub-system* and *buying sub-system*. We denoted these both systems respectively by *Seller_AgS* and *Buyer_AgS*. A buyer mobile agent, denoted by *Buyer_MAg*, will be launched by a *Buyer_AgS*. It will be posted on the Internet to visit several Seller_AgS and find the most interesting offer. Different types of attacks can occur during an electronic transaction. To be secured against these attacks, several security rules may be enforced in a mobile agent-based system. For example, we should enforce a security rule which prohibits any sale transaction at a daily inventory. This rule is specified by a Z schema *SRule_Init*. This schema represents an instance of the *SRule* schema defined in the security framework (Section 3).

$$\begin{array}{|l} BuyerMAg : MobileAgent \\ SellerAgS : AgentSystem \\ R1, dd : Data;\ p : CResource \\ Selling : Action \\ t, t2 : Time \\ \hline p \in ss.Reserved\_res \\ t.hour = 01 \wedge t.minutes = 30 \\ \wedge t.secondes = 25 \\ t2.hour = 02 \wedge t2.minutes = 40 \\ \wedge t2.secondes = 35 \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{SRule\_Init} \\ \hline SRule \\ \hline Name = InterdVente \\ Interested = AgSSellerAgS \\ RSubject = \{MAgBuyerMAg\} \\ Target = \{Rsp\} \\ Type = Prohb \\ Context = Between(t, t2) \\ Actions = \{Selling\} \\ \hline \end{array}$$

We suppose that the inventory is done between 01h 30mn 25s and 02h 40mn 35s. Then, the instance of security rule, presented above, prohibits the *Buyer_MAg* to benefit from resources and services of the *Seller_AgS* at the specified time interval.

Thereby, Whenever the *Buyer_MAg* would buy a product from *Seller_AgS*, the interceptor will check the system time and the type of the security rule instance to decide if the mobile agent is authorized to purchase the product or not. We can also use the dynamic weaving to weave or unweave aspects at runtime in order to insure the adaptability of the security policy further to agent mobility.

To enforce the security rule, we use our security aspect generator according to the syntax of JBoss AOP. Its exploitation allow an automatic generation of different parts of the security aspect. In order to ensure dynamic adaptability of security policies, we use the binding code which allows to weave and unweave aspects at runtime. When the security aspect is weaved into the business code, there will be respect of the security rule. However, when the security aspect is unweaved, there will be execution of the application like if the security aspect doesn't exist without any interception of the aspect.

7 Related Work

Several research works have been proposed for defining security policies for mobile agent systems. Many of them have benefited from the already proposed security frameworks for the management of distributed systems and they integrated them within a mobile agent-based infrastructure. For example, *SPL* (an access control language for security policies) has been used to define, statically, a history-based security policies in mobile agent systems [13]. These policies are enforced by a security monitor, which check event properties and decide about the event acceptability. *Ponder* [14] (a declarative object-oriented policy language) has been integrated within a mobile agent infrastructure in order to ensure security for mobile agents [15]. In this work, authors present a general architecture that provides an automatic mapping of high-level Ponder policy specifications into low-level policies implementation in the Java.

These researches lack an appropriate security framework for mobile agent systems which bring more details to the concepts emerging from combining security with agent mobility and which treats the different concerns of security. Moreover, the researches provide only a system architecture to implement the specified policies in the Java environment without, clearly, explain the mapping from the specification-level to the implementation-level. This is due to the lack of separation between functional concerns and technical concerns of an application.

Other works focus rather on the definition of a security framework specifically adopted to protect mobile agent systems. For example, in [16] the proposed trust framework, which expresses trust and security in mobile agent systems, has not been formally defined. Similarly, the policy based management framework, proposed in [17], describes informally how to protect system-level resources and agents against unauthorized access. The disadvantage about these works is they are devoid of any formal foundations which provide a rigorous reasoning about the consistency of security policies. In fact, these works have studied the specification of policies only at a static level. Moreover, these researches lack a complete view of security aspects in mobile agent systems. Indeed, they specify security policies, only to control mobile agent behaviors and their access resources. Furthermore, agent's representation is generally limited on a simple object deprived of all necessary concepts to express its autonomy and its cognitive aspect. This lack of investigation, is justified by the double complexity bound, on one hand, to the richness and the variety of the concepts for expressing security policies and on the other hand to the richness of the concepts which describe a mobile agent system.

As for the dynamic aspect that is few considered by the community. This aspect reflect the dynamic which characterizes mobile agent systems and the continuous emergence of new security threats in such environments [18]. Many of works that have manipulated this aspect, focused rather on the definition of a security framework specifically adopted to protect mobile agent systems. These works are defined at an architectural level. Consequently they lack formal foundations to reason rigorously about mobile agent security concerns and the dynamic variation of its requirements.

The main difference between all these works and ours is the reliable enforcement of security policies and the ability to manage their reconfiguration at runtime further to agent mobility. This dynamic reconfiguration can't be ensured without a high level of modularity of application. Indeed, we have take advantage of the AOP paradigm, which provides a high-level of modularity, to define a generative aspect-based approach that generates the corresponding security aspects to the reliable specification of security policies.

Several aspect-based approaches have been proposed to enforce security policies. Some of these approaches use aspect-oriented modeling to enforce security policies. Generally, they use UML as modeling language extended by the aspect and weaving concept as proposed in [19]. Other approaches are interested in the security policies only in the implementation level. In this approaches, they use AspectJ as aspect language for enforcing statically security policies. Other approaches use different dynamic aspect weaver, like JBoss AOP in our approach, to enforce dynamically security policies.

8 Conclusion

In this paper, our contribution has focused on the definition of an operational framework for enforcing security policies in mobile agent-based systems. The proposed approach combines aspect oriented programming and formal methods. Our approach consist of three steps:

- The definition of a security aspects template, which is specific to the adopted weaver and based on the generic aspect template.
- The automatic generation of security aspects.
- The dynamic weaving of aspects with the application functional code.

The great benefit of the proposed approach, it is that the consistency proof of security policies avoids any risk of interactions between the generated security aspects. This approach was experimented for securing e-commerce transactions.

As future work, we plan to expand the capacity of our generator to be able to ensure security for other types of constraints and for security rules which has the type *Obligation*. We plan also to adopt a technique to validate the generated code compared to the input specification like the test generation (e.g. JavaCard), assisted proof (e.g. Coq, Isabelle), certified development (e.g. Method B), etc.

References

1. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J., Irwin, J.: Aspect-Oriented Programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
2. Viega, J., Bloch, J.T., Ch, P.: Applying aspect-oriented programming to security. Cutter IT Journal 14, 31–39 (2001)
3. Talhi, C.: Memory-Constrained Security enforcement. PhD thesis, Faculty of Graduated Studies at Laval University, Canada (2007)
4. Erlingsson, U., Schneider, F.B.: SASI enforcement of security policies: A retrospective. In: Proceedings of the 1999 Workshop on New Security Paradigms, pp. 87–95. ACM, New York (1999)
5. Schneider, F.B.: Enforceable security policies. ACM Transactions on Information and System Security 3, 30–50 (2000)
6. Woodcock, J., Davies, J.: Using Z: Specification Refinement and Proof. International Thomson Computer Press (1996)
7. Meisels, I., Saaltink, M.: The Z/EVES Reference Manual (for Version 1.5). Technical report, ORA Canada (1997)
8. Khan, K.: JBoss AOP: Framework for Organizing Cross Cutting Concerns (2006), http://jboss.org/jbossaop/
9. Loulou, M., Kacem, A.H., Jmaiel, M., Mosbah, M.: A Formal Security Framework for Mobile Agent Systems: Specification and Verification. In: Proceedings of the 3rd International Conference on Risks and Security of Internet and Systems, Tozeur, Tunisia, pp. 69–76. IEEE, Los Alamitos (2008)
10. Kallel, S., Charfi, A., Mezini, M., Jmaiel, M., Klose, K.: From Formal Access Control Policies to Runtime Enforcement Aspects. In: Massacci, F., Redwine Jr., S.T., Zannone, N. (eds.) ESSoS 2009. LNCS, vol. 5429, pp. 16–31. Springer, Heidelberg (2009)
11. Greenwood, P., Blair, L.: A framework for policy driven auto-adaptive systems using dynamic framed aspects. In: Rashid, A., Aksit, M. (eds.) Transactions on Aspect-Oriented Software Development II. LNCS, vol. 4242, pp. 30–65. Springer, Heidelberg (2006)
12. Aglets: Mobile Agent System: Aglets (1996), http://www.trl.ibm.com/aglets/
13. Dias, P., Ribeiro, C., Ferreira, P.: Enforcing history-based security policies in mobile agent systems. In: Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks, p. 231. IEEE Computer Society, Los Alamitos (2003)
14. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Sloman, M., Lobo, J., Lupu, E.C. (eds.) POLICY 2001. LNCS, vol. 1995, pp. 18–38. Springer, Heidelberg (2001)
15. Montanari, R., Stefanelli, C., Dulay, N.: Flexible security policies for mobile agent systems. Microprocessors and Microsystems 25, 93–99 (2001)

16. McDonald, J.T., Yasinsac, A.: Application security models for mobile agent systems. Electronic Notes in Theoretical Computer Science 157, 43–59 (2006)
17. Ugurlu, S., Erdogan, N.: A flexible policy architecture for mobile agents. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2006. LNCS, vol. 3831, pp. 538–547. Springer, Heidelberg (2006)
18. Hashii, B., Malabarba, S., Pandey, R., Bishop, M.: Supporting reconfigurable security policies for mobile programs. International Journal of Computer and Telecommunications Netowrking 33, 77–93 (2000)
19. Georg, G., Ray, I., France, R.: Using aspects to design a secure system. In: Proceedings of the Eighth International Conference on Engineering of Complex Computer Systems, p. 117. IEEE Computer Society, Los Alamitos (2002)

Achieving Life-Cycle Compliance of Service-Oriented Architectures: Open Issues and Challenges

Theodoor Scholte[1] and Engin Kirda[2]

[1] SAP Research
805 Avenue du Docteur Maurice Donat
06254 Mougins Cedex France
theodoor.scholte@sap.com
[2] Institut Eurécom
2229, Route des Crêtes
06560 Valbonne, France
kirda@eurecom.fr

Abstract. The introduction of regulations such as the Sarbanes-Oxley act requires companies to ensure that appropriate controls are implemented in their business applications. Implementing and validating compliance measures in 'agile' companies is time consuming, costly, error-prone and a maintenance-intensive task. This paper presents an approach towards dynamically adapting a Service Oriented Architecture (SOA) such that business applications remain compliant. In order to ensure compliance, a compliance checking mechanism for the SOA is needed. Upon detection of a threat/violation, the components of a business application are adapted using aspect-oriented programming (AOP). In this paper, we discuss the fundamental problems and we give an architectural description of our approach.

Keywords: Business Process Management, Compliance Management, Compliance Checking, Service-Oriented Architectures, Aspect-Oriented Programming, Risk Assessment, Risk Mitigation.

1 Introduction

In order to survive in today's business world which is characterized by fact-paced market development, emerging technologies, increased time-to-market pressure and shortened product life cycles, enterprises need to be able to quickly adapt in terms of business processes, partners and relations.

The introduction of regulations such as the Sarbanes-Oxley act [36], Basel II Accord [34], Code Tabaksblad [21], HIPAA [8], IFRS [5], MiFID [7] and LSF [17] requires organisations to implement an effective internal controls system in the enterprise. Non-compliance to rules and regulations can be the cause of juridical pursuits as financial scandals have shown. Examples include Enron and WorldCom in the US and Parmalat in Europe [10,12]. More recently, it became clear that the absence of proper policies, regulations and controls are one

J. Garcia-Alfaro et al. (Eds.): DPM 2009 and SETOP 2009, LNCS 5939, pp. 253–267, 2010.

of the factors that caused the subprime mortgage crisis which resulted in government bailouts of financial firms, bankruptcies or selling of banks at fire sale prices [38]. The term Compliance Management refers to identifying, modeling and implementing rules and regulations such that illegal and illicit behaviour will be avoided when performing business activities. Thus, proper Compliance Management helps in mitigating the risks to illegal, illicit and fraudulent behavior and financial losses. Regulations and legislations constrain the business and are organisation-centric, business-centric, information-centric, legal-aspects centric and human-centric descriptions [26]. They are often imposed by external entities such as the government. Implementing these rules and regulations is difficult as they are documented and communicated in natural language. Furthermore, they are expressed at a high-level of abstraction which means that they have to be translated into executable models and policies such that they can be enforced by the underlying infrastructure. This mapping is always done manually as there are no tools available to automate this process. For these reasons, designing a business process that satisfies laws, rules and legislations and implementing it on top of an IT-infrastructure is a time consuming, costly and error-prone process.

Business applications are used by companies to help them in achieving their business goals. Business goals are reached by performing business activities which can be described by business processes. A popular way to develop a business application is by modeling and implementing a business process through the use of the Service Oriented Architecture (SOA) paradigm. In this paradigm, the functionality of an IT system is structured in small units called services. Then, business processes are modeled to orchestrate these services in order to implement a business activity. The services providing the implementation may change independently from the process specification; enabling and accelerating business & IT alignment and agility. The implementation of a business process requires different stakeholders to be involved due to the increased size and complexity of today's organizations. This issue has been addressed in existing work on Business Process Management (BPM) [18,23,37] and Enterprise Architecture [44].

Compliance Management requires the modeling and the implementation of constraints in the implemented business process. Regulations and business objectives change independently and irregularly from each other. Business applications that are compliant to rules and regulations, are designed and managed through separate activities and by several different experts which have different domain knowledge [22] (e.g. risk and juridical experts). As mentioned above, the mapping of abstract and high-level compliance requirements to implementable rules and policies is a manual process. Therefore, managing compliance is not only time consuming, costly and error-prone but also maintenance-intensive [26]. A scalable, robust and powerful approach is desired to solve the above issues.

In this paper, we make the following contributions:

- We identify the problems related to the semi-automatic adaptation of business applications given a set of constraints that mitigate the risk that illegal/illicit behavior will occur.

– We propose an architecture of an application that can potentially solve the identified problems.

This paper is structured as follows. In section 2 we discuss the problems related to compliance management, section 3 presents a solution architecture that allows adapting business applications automatically based on compliance rules and run-time information. In section 4 we discuss related work. Finally, a summary and an outlook on future research is given in section 5.

2 Problem Discussion

2.1 Case Study

The following use case is used in the EU FP7 project MASTER [9] and it contains the standard business processes that are in use by one of the largest hospitals in Milan. Here, the use case will be used to explain the basic concepts of business processes and internal controls. We use this example to motivate our research problem.

In this real hospital in Milan, drugs are dispensed to patients according to the business process depicted in Figure 1. Modeling a business process for dispensing drugs is normally a complex activity. Therefore, we use a simplified example. The business process starts with a patient who hands a prescription sheet to a doctor or nurse. The doctor/nurse logs into the dispensation software application, the operational unit of the doctor is identifier. Then, the doctor is able to select

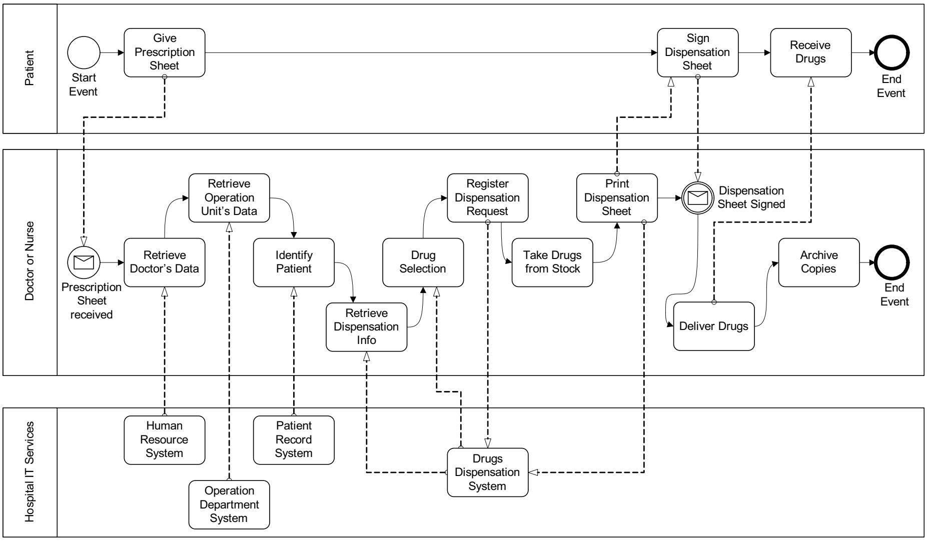


Fig. 1. Drugs dispensation Business Process in use by one of the largest hospitals in Milan

the patient who should receive the drugs. The system receives all the necessary information to select the drugs to be dispensed. The doctor chooses the drugs, registers this, takes the drugs from stock and registers this. Then, the doctor hands the drugs to the patient. The business process contains manual activities as well as activities that are implemented as IT Services. The presence of automated activities are illustrated by the arrows between the 'Doctor or Nurse'-lane and the 'IT Services'-lane. For the rest of this paper we assume that the dispensation software application adopts the SOA paradigm. Thus, the specification of the business process model as shown in Figure 1 is deployed on a business process engine which orchestrates Web Services.

The managers of the hospital wish to make sure that all the medical and non-medical operations that are performed in the hospital conform to a set of relevant internal controls. Examples of relevant compliance rules for the business process depicted in Figure 1 include:

1. Only doctors or nurses are allowed to access the dispensation software application.
2. A doctor/nurse cannot dispense drugs to him or herself (i.e. being a patient and doctor at the same time is not allowed).

The business process depicted in Figure 1 describes *how* a business activity should take place in order to meet a business objective. This is in contrast with compliance rules such as the ones listed above as they are declarative meaning that they indicate *what* the hospital can do in order to satisfy a control. The compliance rules that are listed above typically *imply* certain behaviour and they constrain certain behaviour. The business applications in use by the hospital might or might not be compliant with the compliance rules stated above. Our aim is to adapt the business application automatically when a violation of a compliance rule occurs resulting in compliant business applications and thus, mitigating the risk of additional illegal and illicit behavior occurring.

2.2 Solving Non-compliance

In the previous sections, we have explained what compliance management means and why it is difficult to manage compliance of business applications. We now explain the problems of the existing approaches towards compliance management in more detail.

Companies perform audits to ascertain the validity and reliability of information and to assess to which extent the systems implement compliance measures. Audits are performed in order to be certified as being compliant to certain regulations. The outcome of an audit is a set of risks that are relevant to the organisations assets and an evaluation of the effectiveness of the controls that mitigate risks. The auditing-approach to compliance management generates high-costs as it requires auditors with the necessary expertise and knowledge and audits have to be performed on a regular basis. Audits require experts on regulations as well as experts of the organisation's business and IT-infrastructure to be involved. Due to the complexity, audits check only a part of the business and IT landscape by adopting statistical sampling. The outcome of an audit can be that risks have increased and/or existing compliance measures are not effective enough due to, for example, new versions of regulations being introduced, changes being made in business processes and/or IT-infrastructure. Then, the business processes and the underlying IT-infrastructure need to be adapted such that risks are mitigated. As mentioned in the previous sections, managing compliance is time consuming, costly, error-prone and maintenance-intensive. A software solution that detects violations and potential violations, also called threats, of compliance rules can support the management of compliance. In addition, the software should 'solve' threats and violations by adapting the business process, the business logic and its underlying IT-infrastructure such that non-compliant behaviour will be prevented or compensated, and risks for the organisation are thus mitigated.

In the following sections, we explain the problems that are specific to this software solution.

Modeling behavior and constraints. Business processes and its underlying IT-infrastructure can be described by behavioural models including orchestration models and choreography models. While a choreography model like WS-CDL [40] specifies a collaborative behavior of two or more participants, a business process orchestration model like WS-BPEL [33] specifies a composition of activities designed to achieve a certain business goal. The orchestrations are defined by specifying which services and operations should be invoked. Compliance rules *imply* and *constrain* certain behaviour. We can identify four types of constraints:

- Security constraints
  This type of constraints include all the constraints that have to be put on the system in order to meet security requirements such as confidentiality, integrity, authentication, authorization, availability and non-repudiation. A well-known example of a security constraint is the Segregation of Duties or four-eye principle constraint. This constraint requires multiple persons to complete a task. The second compliance rule in the case study is an example of a Separation of Duty constraint.
- Domain-specific constraints
  These constraints refer to the business rules of the enterprise and are specific to the context/domain of the enterprise. An example in the context of drugs dispensation could be: when a patient gets Paracetamol and a Blood Thinner dispensed, a warning should be raised.
- Orchestration constraints
  Orchestration constraints include dependencies between the activities in a business process and are specified in a business process model. An example based on the scenario in section 2.1 is that a 'Take Drugs from Stock' activity must be followed by a 'Print Dispensation Sheet' activity.
- Choreography constraints
  This type of constraints are the ones that are enforced over the interaction between business partners and are specified in a choreography model. Consider a business process where doctors prescribe drugs and a pharmacy dispense

> drugs. A doctor can only send a prescription to the pharmacy if he received an acknowledgement of the previous prescription from the pharmacy.

Please note that this classification does not enforce that a particular constraint falls within one specific class of constraints. For example, a constraint such as ‘a patient should not get Paracetamol and Blood Thinner dispensed at the same time’ is a domain-specific constraint. But it is also an orchestration constraint when the orchestration model specifies an activity ‘check dangerous drugs combinations’ followed by a ‘warn doctor’ activity. The verification of the implementation of a business process requires a language to describe the model of the target system and a language to describe the constraints that have been put on the system. This language should be expressive enough to model the semantics of the rules and regulations that exist in the real-world, and yet abstract enough for the purpose of validation and analysis.

Detecting threats and violations. Threats and violations have to be detected during the whole lifecycle of business processes. This can be achieved by compliance checking which refers to the verification of the status of compliance measures in the enterprise [15]. We can identify two complementary approaches for compliance checking: design-time and run-time. Compliance checking at design-time means the verification of a behavioural model (formal model) of the implemented business process against a set of formally specified constraints (compliance rules). The behavioural model is compliant if its definition complies with the predefined set of compliance rules. Runtime compliance checking is based on the evidence collected at runtime and is required here to detect whether threats or violations of constraints occurred in practice. With respect to orchestration and choreography constraints, verifying the behavioral models (at design-time) is possible but it is not sufficient enough as unexpected behavior might occur at runtime. In addition, the validation of security and (data) integrity constraints requires information that is not available at design-time. The challenge for compliance checking, is to come up with an approach that does not only detect non-compliant behaviour but is also able to trace back to the causes of non-compliant behaviour.

The problem of adaptation. When threats and/or violations of compliance rules have been detected, business applications should be repaired automatically such that violations can be prevented or violations can be compensated and risks are mitigated. Software adaptation can either be done at *design-time* or at *runtime* [6]. Design-time adaptation refers to all types of changes made to software before the software system is running. This can include modifications of requirements/specifications, modifications of source code or changes of configuration files. An important property of this type of software adaptation is that all the steps in the adaptation process are known and have been planned in advance. This is in contrast to dynamic or runtime software adaptation that refers to techniques that allow to change running pieces of software. In the context of business process driven applications, adaptations can be made by modifying the business process model. A business process can be modified by inserting, deleting or shifting activities in the service orchestration [11]. Just as with generic software adaptation, modifications of the orchestration and/or choreography model can be applied at design-time or at runtime. Design-time adaptation is needed when the business application has to meet new (compliance) requirements and there is an intention to reuse the modifications. Dynamic or runtime adaptation allows us to bring a running business application to a compliant state without the need to restart the application which might result in loss of data. Since we would like to prevent and compensate violations before the execution of a business application is finished, dynamic software adaptation techniques should be applied. An important issue here is maintaining the consistency of the control flow and of runtime data. This requires the development of adaptive middleware. A second issue with respect to dynamic adaption is that the compliance rules do not specify *how* to repair non-compliant behaviour. Depending on the business process, its implementation and a compliance rule, there might be more than one way or *strategy* to repair a business application. Thus, a compliance rule is always associated with one or more repair strategies. The challenge here is the modeling and coding of these repair strategies.

3 System Overview

Figure 2 depicts the global overview of our system. Business process models define the way how Web Services are orchestrated and these models are deployed on a BPEL engine [33]. In addition, the system includes an XACML policy engine [32] which is responsible for evaluating access control requests originating from Web Services against a set of access control policies. The adaptors that are responsible for collecting evidence of software behavior and the adaptors for repairing the software are implemented using aspect-oriented programming (AOP). The main reason for choosing an AOP approach is that it supports *compile-time, load-time and runtime weaving* [35,39].

Aspect-Oriented Programming and Annotations. The following paragraphs gives a very short introduction on AOP and annotating source code. For a more comprehensive introduction, consider reading [24,25] and [4] for source code annotations.

Aspect-oriented programming is a programming paradigm which aims to provide modularizing techniques supporting the separation of cross-cutting concerns in complex software systems. Examples of cross-cutting concerns include security constraints, logging functionality and communication protocols. The main idea is to separate the cross-cutting concerns in stand-alone modules called *aspects*. An aspect is related to one or more places in the code which are called *join points*. In order to identify join points, the notion of a *pointcut* is introduced. The additional behavior at a join point is specified in an *advice* and this code can run before, around or after a join point. The AOP framework is responsible for combining the base functionality with the additional code, this step is called weaving. Weaving can be done at compile-time (by the compiler), load-time (by

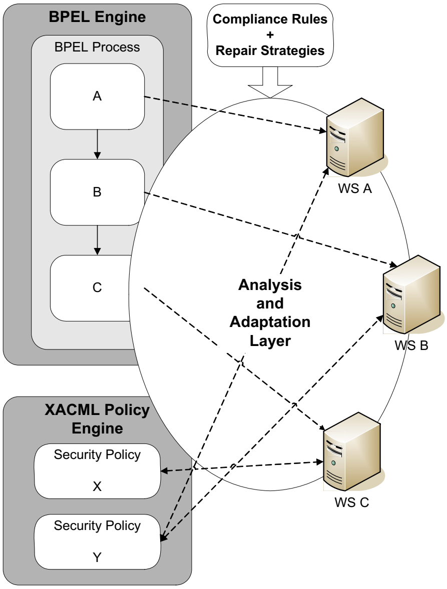


Fig. 2. High-level overview of an adaptive Service-Oriented Architecture

the classloader) or at runtime. With runtime weaving, targets can be declared at runtime. The class bytecode can be redefined at runtime without reloading the class.

Annotations, and in particular Java annotations, are a special form of metadata that can be added to Java source code. All types of declarations can be annotated: packages, classes, variables and methods. Unlike javadoc, Java annotations may be available at runtime. The Java VM may retain annotations and make them retrievable at runtime. Annotations do not directly affect the application semantics, but they can be processed by tools at design-time or at runtime. Then, these tools can affect the application behavior. A similar concept exists for .NET.

3.1 Architecture

Figure 3 depicts the runtime architecture of the approach taken. Due to space constraints, we left out the architecture of the design-time infrastructure. The compliance monitoring and adaptation process starts with modeling compliance rules and repair strategies using a modeling tool. These models are stored in a repository (1). Whenever a new compliance rule or repair strategy is added to the repository, the coordinator (2) is notified and translates the compliance rule to queries for the probing infrastructure (7), the monitoring infrastructure (4) and the compliance analyzer (3). The probing infrastructure (7) is responsible for collecting evidence of the system's behavior. Each probe is a 'hook' into a

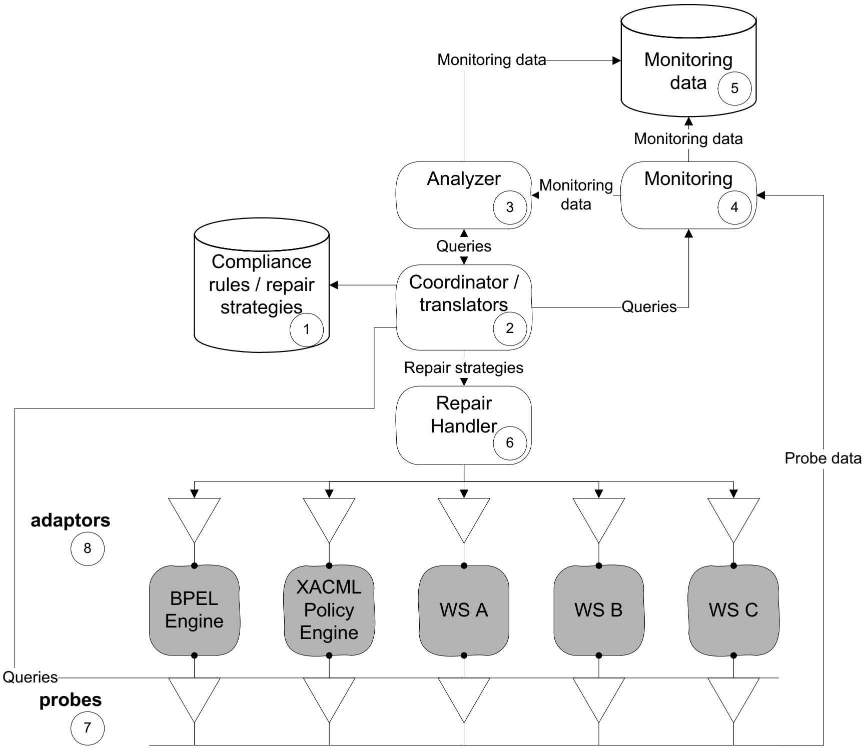


Fig. 3. Detailed run-time architecture of an adaptation system for a Service-Oriented Architecture

component of the Service-Oriented Architecture (BPEL engine, policy engine, Web Service) and emits events that represent the behavior of a component. In order to reduce the number of events that are emitted by the probes and to 'enrich' the evidence, there is a monitoring component (4) which is implemented by a *complex event processor*. This component filters, aggregates and correlates events. The monitoring component fills a repository (5) which contains historical data of the actions that were performed by the Service-Oriented Architecture. The compliance analyzer (3) analyzes continuously the historical evidence from the repository (5) and the accurate evidence originating from the monitoring infrastructure (4). If the compliance analyzer detects a violation of a compliance rule while analyzing the evidence, the coordinator (2) is notified. The coordinator retrieves the corresponding repair strategies (advice) from the repository (1) and sends them to the repair handler (6) together with the location of the source of the evidence. Based on this location, the repair handler chooses the locations of the adaptation. This location is composed of a component identifier and a pointcut which identifies the join point to insert the additional code. The component identifier identifies the adaptor (8) to which the repair handler should send the advice.

Probes and Adaptors. The probes and adaptors are both implemented using the aspect-oriented programming (AOP) paradigm. These components are added to the base functionality of the Service-Oriented Architecture as advices. The main reason is that this approach allows us to enable or disable a probe or adaptor (and the adaptor's associated repair strategy) at compile-time, load-time or runtime. The coordinator determines *where* in the target system to put probes and the repair handler determines this for the adaptors. The coordinator and repair handler specifies this in a pointcut. Pointcuts can be specified using a language which syntax is based on the base language, like Java signatures when Java is used. In order to determine and define the crucial parts of a target's system component where evidence collection and adaptation should take place, a more fine-grained way of specifying point cuts is needed. In our approach, we annotate the source code of each component in the target system and we expose the annotations to the coordinator and the repair handler. By referring to the annotations in the pointcuts, the coordinator and the repair handler are able to determine and define the locations for evidence collection and adaptation.

3.2 Case Study Revisited

In this section, we describe the behavior of our Compliance Management Solution when a violation of the compliance rule 'Only doctors or nurses are allowed to access the dispensation software application' (section 2.1) occurs. This rule implies that every Web Service implements an access control mechanism. As mentioned above, we assume that there is a centralized access control policy engine (Policy Decision Point in XACML terminology) and every Web Service (Policy Enforcement Point) implementing an activity of the business process depicted in figure 1 sends requests to this policy engine. Now, the compliance rule can be refined to the following compliance rule 'always when a Web Service needs access control the PDP has to receive an access control request message'. This rule can be expressed in LTL as follows:

```
[]((Q & !R & <>R) -> (P U R))
```

where Q *refers to the event representing the need for access control,* P *refers to the event representing the access control request message at the PDP and* R *is a time-bound equal to* $\mathsf{Q} + 30$ *seconds*

This LTL property is stored in the 'compliance rules / repair strategies' repository together with a repair strategy (advice) that includes all the source code for making an XACML request, sending it to the policy decision point and processing the response. An XACML request requires parameters such as subject's and resource' identity and the action. We assume that the variable-names of these parameters can be found in the annotations. Then, the repair handler can retrieve them and generate the correct repair strategy. Due to space constraints, we do not give an example here of the source code of a XACML request. When the combination of a compliance rule / repair strategy is added to the repository, the coordinator gets notified and translates the compliance rule to queries for the probing infrastructure, the monitoring components and the compliance analyzer. Consider now that a user performs the activity ‘Register Dispensation Request’ of the business process depicted in figure 1. The Web Service implementing this activity starts executing the method RegisterDispensation. The coordinator identified this as a critical point (based on the annotation and compliance rule) and put in advice in place that emits an event when this method is called. The event represents the need for access control. The XACML policy engine does not emit an event representing an access control request. The compliance analyzer detects this as there is a violation of the compliance rule. The coordinator passes the source of the violation of the compliance rule to the repair handler together with the repair handler. The repair handler concludes that the web service does not implement a valid access control request mechanism and decides to apply the repair strategy that performs a XACML request to the centralized policy engine.

4 Related Work

Related work in the context of compliance focuses mainly on modeling controls and compliance validation but not on software adaptation techniques.

Compliance modeling. The ability to model compliance rules are essential for our solution. Models of compliance rules are stored in the repository and the coordinator translates them to queries that can be deployed on the evidence collection and processing infrastructure. The control pattern introduced in [30,31] acts as a pattern-based abstraction layer that separates business process and compliance management by annotating the process model with compliance rules. The approach is promising because the patterns have been used for run-time compliance validation. However, it lacks support for modeling constraints between process instances and modeling context. Moreover, the applicability of control patterns for dynamic adaptation of SOAs has not been shown. In [41,42], Wolter et al. propose to use annotated business process models to model security requirements. The approach allows to extract security policies such as AXIS2, XACML and WS-Policy security configurations from an annotated business process model. However, compliance rules include more than only security requirements such as authorization, access control and encryption.

Design-time compliance checking. Approaches towards a priori or design-time compliance checking are based on the concept of validating a specification of a process model against a certain set of compliance properties including the ordering of activities, liveness and correctness properties. Although our approach does not include design-time compliance checking, we describe here some work in that area. The approaches proposed in [3,13,16,22,27,43] are all based on a priori compliance checking. The differences between the approaches are: 1) the languages used to specify the process models and the compliance properties 2) the model checker or reasoning techniques used. In [13], Concurrent Transaction Logic (CTR) is used as the language to specify, analyze and to schedule workflows. The compliance properties or the constraints are specified as CTR formulas and also workflow graphs are transformed to CTR formulas. Liu et al. proposed in [27] a compliance-checking framework that allows to model process models in BPEL and compliance properties in the graphical Business Property Specification Language (BPSL). Model transformation techniques are used to map the BPEL process models to FSMs and the compliance properties to LTL properties. [3] does not use BPEL but BPMN diagrams which are translated in REO models. The approach presented in [16] focuses on verifying the compliance of service interactions against obligation policies. These policies describe what actions a subject must or must not do to a set of target objects. The service interactions are specified in BPEL, the obligations in Message Sequence Charts (MSC). A similar approach is [43] in which a BPEL process is validated against properties expressed using property patterns [14].

Runtime compliance checking. In our solution architecture, run-time compliance checking is used to detect the cause of non-compliant behaviour in a SOA. In [19,20] a method and meta-model is introduced that captures compliance requirements in a language. Using this framework, abstract policies can be translated to implementation artifacts such as business process definitions, data retention policies, access control lists and monitoring policies. This model-transformation process can, at least partially, be carried out automatically. Agrawal et al. addressed in [1] the importance of using database technology for run-time and a-posteriori compliance checking. The work in [2] focuses on an event-based language that can be used for run-time monitoring of Web Service interactions. The work presented in [15,29,28] propose different frameworks for compliance management. All of them adopt both design-time and run-time compliance checking techniques. The work looks only at the level of business process execution while we are planning to look at a lower-level. In [30,31], a semantic mirror is used to collect run-time information of process instances. Violations to pre-defined control patterns are detected by the semantic mirror.

5 Conclusion and Outlook

The introduction of laws and regulations leads to the need to identify, model and implement proper controls in the IT-landscapes of organisations such that illegal and illicit behaviour can be avoided when performing business activities. Managing compliance in ‘agile’ companies requires the use of a software solution that is able to detect non-compliant behaviour and adapts the components of business applications accordingly. In this paper, we identified the problems this software solution should cope with and we presented an architecture which uses the aspect-oriented programming paradigm for evidence collection and software adaptation. Future work will focus on developing a proof-of-concept of the proposed architecture. Moreover, one of the open issues is determining the relevant join points for evidence collection and adaptation. We gave some hints and directions, but additional research is necessary. Another open issue is the translation from compliance rules to queries that can be evaluated on probes, the monitoring infrastructure and the compliance analyzer.

Acknowledgements. This work was supported by funds from the European Commission (contract N 216917 for the FP7-ICT-2007-1 project MASTER).

References

1. Agrawal, R., Johnson, C., Kiernan, J., Leymann, F.: Taming compliance with sarbanes-oxley internal controls using database technology. In: ICDE 2006: Proceedings of the 22nd International Conference on Data Engineering, Washington, DC, USA, p. 92. IEEE Computer Society Press, Los Alamitos (2006)
2. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Montali, M., Storari, S., Torroni, P.: Computational logic for run-time verification of web services choreographies: Exploiting the *ocs-si* tool. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 58–72. Springer, Heidelberg (2006)
3. Arbab, F., Kokash, N., Meng, S.: Towards using reo for compliance-aware business process modeling. In: Margaria, T., Steffen, B. (eds.) ISoLA. Communications in Computer and Information Science, vol. 17, pp. 108–123. Springer, Heidelberg (2008)
4. Austin, C.: J2se 5.0 in a nutshell
5. International Accounting Standards Board. International accounting standard 1: Presentation of financial statements
6. Canal, C., Murillo, J.M., Poizat, P.: Software adaptation 14(13), 2107–2109 (2008)
7. European Commission. Markets in financial instruments directive
8. United States Congress. Health insurance portability and accountability act of (1996)
9. EU FP7 MASTER Consortium. Managing assurance, security and trust for services, http://www.master-fp7.eu
10. Creswell, J.: Citigroup agrees to pay 2 billion in enron scandal. The New York Times (June 2005)
11. Dadam, P., Reichert, M.: The adept project: A decade of research and development for robust and flexible process support - challenges and achievements. Computer Science - Research and Development (23), 81–97 (2009)
12. Dash, E.: Parmalat sues citigroup over transactions. The New York Times (July 2004)
13. Davulcu, H., Kifer, M., Ramakrishnan, C.R., Ramakrishnan, I.V.: Logic based modeling and analysis of workflows. In: PODS 1998: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 25–33. ACM, New York (1998)
14. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. Technical report, Amherst, MA, USA (1998)
15. El Kharbili, M., Stein, S., Markovic, I., Pulvermüller, E.: Towards a framework for semantic business process compliance management. In: Proceedings of the First International Workshop on Governance, Risk and Compliance (GRCIS), Montpellier, France, June 17 (2008)
16. Foster, H., Uchitel, S., Magee, J., Kramer, J.: Model-based analysis of obligations in web service choreography. In: AICT-ICIW 2006: Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, Washington, DC, USA, p. 149. IEEE Computer Society Press, Los Alamitos (2006)
17. Gouvernement Francais. La loi de sÉcuritÉ financiÉre

18. Giaglis, G.M.: A taxonomy of business process modeling and information systems modeling techniques. International Journal of Flexible Manufacturing Systems 13(2), 209–228 (2001)
19. Giblin, C., Liu, A.Y., Müller, S., Pfitzmann, B., Zhou, X.: Regulations expressed as logical models (realm). Technical Report RZ 3616, IBM Research, Zurich (July 2005)
20. Giblin, C., Müller, S., Pfitzmann, B.: From regulatory policies to event monitoring rules: Towards model-driven compliance automation. Technical Report RZ 3662, IBM Research (2006)
21. Commissie Corporate Governance. De nederlandse corporate governance code: Beginselen van deugdelijk ondernemingsbestuur en best practice bepalingen
22. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: EDOC 2006: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference, Washington, DC, USA, pp. 221–232. IEEE Computer Society Press, Los Alamitos (2006)
23. Ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
24. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.G.: An overview of aspectJ. In: Knudsen, J.L. (ed.) ECOOP 2001. LNCS, vol. 2072, pp. 327–353. Springer, Heidelberg (2001)
25. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
26. Lang, U., Schreiner, R.: Managing business compliance using model-driven security management. In: Proceeedings of ISSE 2008 Securing Electronic Business Processes (2008)
27. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Syst. J. 46(2), 335–361 (2007)
28. Ly, L.T., Göser, K., Rinderle-Ma, S., Dadam, P.: Compliance of semantic constraints - a requirements analysis for process management systems. In: Proc. 1st Int'l Workshop on Governance, Risk and Compliance - Applications in Information Systems (GRCIS 2008), Montpellier, France (2008)
29. Ly, L.T., Rinderle, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. Data Knowl. Eng. 64(1), 3–23 (2008)
30. Namiri, K., Stojanovic, N.: A formal approach for internal controls compliance in business processes. In: Proceedings of the 8th Workshop on Business Process Modeling, Development, and Support, Trondheim, Norway (2007)
31. Namiri, K., Stojanovic, N.: Pattern-based design and validation of business process compliance. In: On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, pp. 59–76. Springer, Heidelberg (2007)
32. OASIS. extensible access control markup language (xacml) version 2.0 (February 2005)
33. OASIS. Web services business process execution language (2007)
34. Basel Committee on Banking Supervision. International convergence of capital measurement and capital standards: A revised framework
35. Popovici, A., Gross, T., Alonso, G.: Dynamic weaving for aspect-oriented programming. In: AOSD 2002: Proceedings of the 1st international conference on Aspect-oriented software development, pp. 141–147. ACM, New York (2002)

36. Sarbanes, P., Oxley, M.: Sarbanes-oxley act of 2002 (pub.l. 107-204, 116 stat. 745)
37. Sedera, W., Gable, G.G., Rosemann, M., Smyth, R.W.: A success model for business process modeling: findings from a multiple case study (2004)
38. Streitfeld, D., Morgenson, G.: Building flawed american dreams. The New York Times (October 2008)
39. Vasseur, A.: Dynamic aop and runtimeweaving for java - how does aspectwerkz address it? In: Workshop on Dynamic AOP (2004)
40. W3C. Web services choreography description language version 1.0
41. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. Journal of Systems Architecture, 13 (2008)
42. Wolter, C., Schaad, A., Meinel, C.: A transformation approach for security enhanced business processes. In: Proc. SE 2008 of 26th IASTED International Multi-Conference (February 2008)
43. Yu, J., Manh, T.P., Han, J., Jin, Y., Han, Y., Wang, J.: Pattern based property specification and verification for service composition. In: Aberer, K., Peng, Z., Rundensteiner, E.A., Zhang, Y., Li, X. (eds.) WISE 2006. LNCS, vol. 4255, pp. 156–168. Springer, Heidelberg (2006)
44. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. 26(3), 276–292 (1987)

Author Index