European Context-Awareness and Trust (EUROCAT09)

3^{rd} Workshop on Combining Context with Trust, Security, and Privacy

co-located with EUROPKI09 and ISC09
9^{th} September 2009, Pisa, Italy

Workshop Proceedings

Editors

Gabriele Lenzini — Novay, The Netherlands
Marinella Petrocchi — IIT-CNR, Italy

Preface

This volume contains the proceedings of the European Workshop on Combining Context with Trust, Security and Privacy (EUROCAT09), held the 9th September 2009 in Pisa, Italy. The workshop runs its third edition, and it has followed the previous CAT07 and CAT08. The prefix "EURO" underlines the European scope of this edition of the workshop.

In its previous editions the workshop was co-located with IFIP Conference on Trust Management (IFIPTM). The interaction with the trust management community has brought stimulating ideas and experiences from the trust management research into the context-awareness areas. This year the workshop has tried to widen its scientific interaction and it has been co-located with two important events in Security and Privacy: (a) EUROPKI09, the European Workshop on Public Key Services Application and Infrastructures, and (b) ISC09, the Information Security Conference. This change is consistent with the goal of the workshop, which aims to stimulate an active exchange of new ideas on the bidirectional relationship between the area of context awareness and the area of trust, security, and privacy.

Since its very foundation, the workshop had the target of bringing experts together, to collect the state of the art, to identify open and emerging problems in the evolution of today's communication systems. In fact, society is moving fast toward a more pervasive and ubiquitous context-awareness infrastructures (*e.g.*, think about the increasing number of applications that are able to understand what a user is doing, or about the arising of living lab initiatives across Europe) that challenge trust, security, and privacy of individuals and societies. In the same time, the use of context may arise new opportunities, for example, in service customization and in application's efficiency and user friendliness.

The focus on combining context with trust, security, and privacy open new insights as well as provide an interesting playground for the communities of researchers from Trust, Security, and Privacy. In particular:

- The opportunity to use context as an approach to enhance privacy and security seems an interesting, innovative, and value-adding extension. For example, the availability of context information can help the establishment of trust relationship (users in the same room for the same meeting are more willing to trust each other more than users without visible contact). Moreover, contextual information can be used to improve dynamic, adaptive, and autonomic aspects of security, access control, and privacy enhancing technologies.
- The opportunity to apply the results achieved in security, privacy, and trust to strengthen context aware infrastructures and to facilitate the exchange of context information is a challenge for the Security community. For example, because context information often has a personal character, privacy and

other rights of individuals are potentially endangered and need scientific and technological solution that carefully protect them

Following the trend initiated last year, EUROCAT09 has attracted the attention of researchers with a computer science and information & communication technology background, whose experiences root both in the academia and industry. In the current edition, four papers have been accepted for oral presentations. Papers were reviewed by at least three reviewers of the program committee, and all selected papers are of high quality. The volume opens with a resume of the invited speaker's talk on trust requirements and policy management issues in the design of context-aware and service-oriented architectures which make use of privacy sensitive end-users' information collected from sensors and information providers. The following articles in the volume are representative of the current research activities on the topics of the workshop. Two contributions approach the design of user-centric architecture for identity management and authentication in smart and pervasive ambients. In particular the first focuses on the problem of delegation of identities in ubiquitous environments, the second on how to realize a persistent authentication solution in smart environments where privacy must be also preserved. A third contribution addresses the problem of managing the retrieval of context-aware point-of-interests while preserving user' privacy. The last paper proposes a mechanism that allows a secure portability of entities' reputations across different web-portals.

An on-line version of the present proceedings will be published by CEUR-WS.org (http://ceur-ws.org). It will also be available at the workshop website.

We would like to take this opportunity to thank people who contributed to the EuroCAT09 workshop. We wish to thank the PC members, the reviewers, the invited speaker and, in particular, all the authors for their valuable contributions: we wish them a successful continuation of their work in this area. We thank the chairs of EuroPKI09 and ISC09 conferences for their support and hospitality.

September 18, 2009
Workshop Chairs:
Gabriele Lenzini
Marinella Petrocchi

Organization

EUROCAT09 has been supported by Novay, Enschede, The Netherlands (www.novay.nl) and by the Instituto di Informatica e Telematica IIT-CNR, Pisa, Italy (www.iit.cnr.it).

Chairs

Gabriele Lenzini	(Novay, The Netherlands)
Marinella Petrocchi	(IIT-CNR, Italy)

Program Commitee

Rubén Alonso	(Visual Tools, Spain)
Benjamin Aziz	(STFC, UK)
Noria Foukia	(Univ. Otago, New Zealand)
Paolo Mori	(IIT-CNR, Italy)
Daniel Olmedilla	(Telefónica R&D, Spain)
Daniele Quercia	(MIT, USA)
Daniele Riboni	(Univ. Milano, Italy)
Marco Luca Sbodio	(HP Italy Innovation Center, Italy)
Jean-Marc Seigneur	(Univ. Geneva and Venyo, Switzerland)
Daniel Schreckling	(Univ. Passau, Germany)
Anna Cinzia Squicciarini	(The Pennsylvania State Univ., USA)
Zheng Yan	(Nokia Research Center, Finland)

Table of Contents

Trust Management in Context-Aware and Service-Oriented Architectures (Invited Talk)

Ricardo Neisse

Fraunhofer-Institute for Experimental Software Engineering (IESE)

Extended Abstract

In service-oriented architectures, services are the basic building blocks to dynamically compose complex business process across multiple administrative domains. The main goal is to support companies in the outsourcing of services to service providers that best suit their business needs, and dynamically reassign the services to other providers when changes in the business are necessary. The dynamic re-assignment of service providers in an open service market will only be successful if appropriate trust management mechanisms are put in place to provide guarantees that the desired service requirements are fulfilled. In context-aware and service-oriented architectures, there are additional trust requirements, because this type of service-oriented architecture makes use of privacy sensitive end-users' information collected from sensors and information providers surrounding the end users' physical space.

In this talk, I will report on trust and policy management issues in pervasive and service-oriented architectures. I will briefly discuss the social and legal requirements, describe the trust and policy management challenges we have identified, and introduce our trust and policy management approach to support end-users and service consumers in this service scenario. I will also comment on the ongoing research in the area of trustworthy enforcement and management of policies using the support provided by the Trusted Computing Platform.

Keyworks: Context, trust, policy, management, privacy, pervasive systems, service-oriented architectures.

Context-Aware Identity Delegation

Naveed Ahmed and Christian D. Jensen

Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Lyngby
Denmark
nahm@kth.se,cdj@imm.dtu.dk

Abstract. In emerging ubiquitous computing, related nomadic users often perform similar tasks and share the same computing infrastructure. This means that security of the shared resources is of prime importance. Frequent delegation of tasks among users must be anticipated as most nomadic environments are hectic and very dynamic. A delegation mechanism with a slightly complicated user interface will not only reduce the productivity but also provide nomadic users with a strong motivation to circumvent the mechanism itself.
Delegation in access control domain is not practical for the most of nomadic users due to its complicated and complex structure. Identity delegation at authentication level provides improved usability, which reduces the risk of circumventing the delegation mechanism; at the same time, however, identity delegation violates the principle of least privileges. We use contextual information of a delegatee to mitigate this violation, which helps to achieve a higher level of practical security in nomadic environments.
Keywords: Context-aware, Identity Delegation, Nomadic User, Practical Security

1 Introduction

The pervasive use of computing technology in our life has caused a shift in how we interact with computers. Earlier, in the age of mainframes and desktops, a user had to physically move to a computer to access information or computing resources. When technology allowed manufacturing of lightweight devices along with a well connected wireless communication infrastructure, the paradigm of mobile computing emerged. This allows people to move freely in an environment along with their computing devices, so information and (modest) computing resources are ubiquitously available. More recently, we have started embedding computing devices into work environments, which makes the nomadic use of computing possible. Nomadic users move freely in their work environment and use shared devices that are embedded in the environment on which they can invoke their unique sessions. A typical example is a hospital, where computers and equipment are shared among doctors and nurses. These people access patients' health-care data from different terminals in wards using their unique identities.

In collaborative work environments, nomadic users often need to work in each other's place, for instance, in the nomadic environment of hospitals, a senior doctor may need to delegate his duties to another doctor when a patient in a critical condition arrives in the emergency unit. In this case, the senior doctor is primarily concerned with two things: to *whom* should he delegate and *which* of the patients and wards need to be taken care of. In the hectic environment of current hospitals, doctors generally prefer to simply share their passwords and sessions for the delegation [5], rather than doing so by a complicated procedure that may offer a higher level of security. For these nomadic users, it does not make sense to engage them beyond their simple concerns of Whom and Which for the purpose of delegation; otherwise, they would tend to circumvent the delegation mechanism due to its usability issues.

From a user's perspective, delegation is only concerned with the two simple questions: to *whom* one should delegate, the delegatee; and in *which* context the delegatee should be authorized to work on one's behalf. From a security perspective, however, it is also accompanied with fine level details of individual authorizations, in order to follow the principle of least privileges [3, 15]. A pure security perspective represents one extreme, where a delegation mechanism has fine granularity and thus each privilege is transferred individually, however, this fine control also makes it cumbersome and error prone. Considering the usability perspective, delegation becomes more and more coarse grained, where many privileges are transferred simultaneously, at the expense of the principle of least privileges, but its simplicity could make it more user-friendly and thus more likely to be used in practice.

Delegation is necessarily regulated by a security policy, which defines the privileges of each user and their authority to delegate them. We call this the *Designed Security Level*. The designed security level is not an actual assessment of the security of the system, but captures the intentions of the designers through their specification of policies and choice of mechanisms. However, the security that we achieve in practice is often less than or at most equal to the designed level, we call this the *Effective Security Level* because security mechanisms may contain vulnerabilities and users may decide to circumvent the mechanism, e.g., by sharing passwords. Most systems are designed to achieve a specific level of security, which is perhaps mandated by legislation, so the designed security level tends to remain constant over time. The effective security level, however, tends to degrade with time as vulnerabilities are discovered – but not necessarily patched, advances in technology makes new attacks possible, new techniques in cryptanalysis are discovered or simply that the vigilance of users erode and shared passwords proliferate in the system. This increasing *security gap* between the designed and effective security levels is illustrated in Figure 1.

It is important to note, however, that the curve shown in the figure is for illustrative purposes and exact shape of the curve is not relevant; the actual shape depends on the type of the system, the configuration and operation of the system, and parameters of the computational environment in which the system is deployed. For example, new security mechanisms, for which the users are not

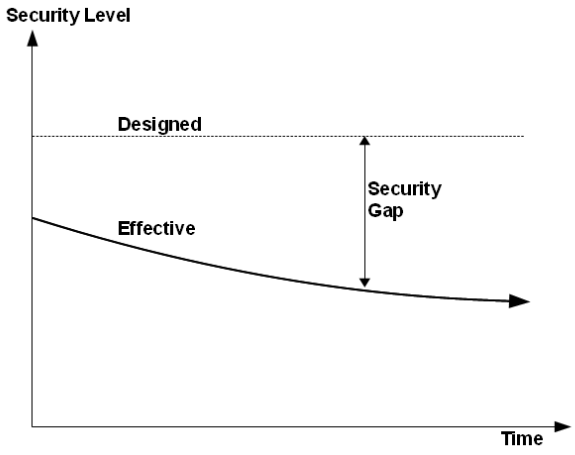


Fig. 1. Difference between Effective and Designed Security Levels

yet trained or aware of its purpose, the curve of effective security could have the positive slopes in the start. In any case, the poor usability in delegation provides a motivation to users for deceiving the access control mechanism by sharing their authentication tokens, which also contributes to the widening security gap.

The security gap could be reduced by improving the usability of delegation. The identity delegation at authentication level has a considerable usability advantage over classic delegation models, but on the expense of violating the principle of least privileges [2]. Still, the identity delegation could provide more effective security than classic delegation mechanisms for many nomadic environments where there are pre-established trust relationships among users, such as in a group of doctors who work together in a hospital. We extend the identity delegation [2] to include contextual information, which further increase the usability while at the same time provides more justifications to use the identity delegation despite of its obvious violation of the principle of least privileges. The use of context limits the unnecessary spread of authorizations. We have implemented the proposed mechanism in form of a prototype.

In the next section, we describe part of the state of the art in delegation. Section 3 presents our delegation mechanism in detail. In Section 4, we describe the details of the prototype. Section 5 provides evaluation of the mechanism. In the final section, we present some conclusions.

2 Related Work

The term *delegation* has many definitions, for instance Abadi et al. [1], Barka and Sandhu [6], Gasser and McDermott [9], Gladney [10], etc. In the following, we adhere to the definition of Zhang et al. [19], i.e., we consider authentication

as a process in which one active entity in a system transfers its authority to another active entity in order for that entity to carry out a job on behalf of the first entity. In this paper, we only focus on human-to-human delegation and consider a *delegator* who delegates his authorizations, while one who receives these authorizations is referred as a *delegatee*. [3]

Human-to-human delegation can be at access control level or at authentication level. At the access control level, Gasser and McDermott [9] propose a delegation technique with cryptographic assurances of the revocation if the system is subsequently compromised. Varadharajan et al. [16] consider delegation in distributed systems as the problem of verification for a delegatee using signature-based scheme with certain assumptions of trust relationships among the system entities. Zhang et al. [19] propose a rule-based specification language and a rule-based framework for the role-based multi-step delegation. The delegation at permission level, although with very limited options, can also be accomplished in the UNIX access control model. A formal framework for delegation under Role Based Access Control(RBAC) is proposed by Barka [6]. Bertino et al. [7] presents the notion of context in form of Temporal- RBAC that supports periodic role enabling and the notion of time in form of temporal dependencies among permissions. Covington et al. [8] extends the context of model beyond time by incorporating constraints of location and system status. Kumar et al. [4] presents a formal context-sensitive RBAC model, which enables complex security policies using the context information.

At user's authentication level, an identity delegation essentially assigns the system identifier of a delegator to a delegatee. A framework at this level is invented by Mercredi and Frey [14], whose primary objective is to enable a person to sign in on the behalf of another person. Ahmed and Jensen [2] propose a simpler architecture of identity delegation, which is derived from the usability factors of delegation in nomadic environments. Using *sudo* or *su* commands of UNIX are implementation dependent alternatives.

An identity delegation transfers all authorizations of a user most of which might not be required for the delegated job. Nevertheless, an identity delegation can be more user friendly as all the necessary privileges are delegated in one fell swoop along with the identity, while at access control level, it is usually hard for a user to figure out the exact privileges necessary for performing a particular job. Li and Wang [13] address this problem in access control domain, by bundling the authorizations of a job in form of a unit. On the other hand, revocation of delegation is equally important and might be non-trivial [17].

In nomadic use of computing, where users frequently delegate to one another, complexity of delegation (and its revocation) results in poor usability and provides a strong incentive for users to bypass the secure way of delegation. As indicated by Bardram, et al. [5], users start sharing their authentication tokens for mutual collaboration. On the other hand, the existing user-friendly identity delegation techniques [14, 2] do not consider the context and thus cause a wider spread of authorities, which restricts their use to a limited number of nomadic environments.

3 The Mechanism for Context-Aware Identity Delegation

In the following discussion, the term *Validated identity* refers to the identity that an authentication mechanism concludes with help of one or more authentication techniques. Similarly, the so-called authenticated identity provided to an access control mechanism is referred as *Effective identity.*

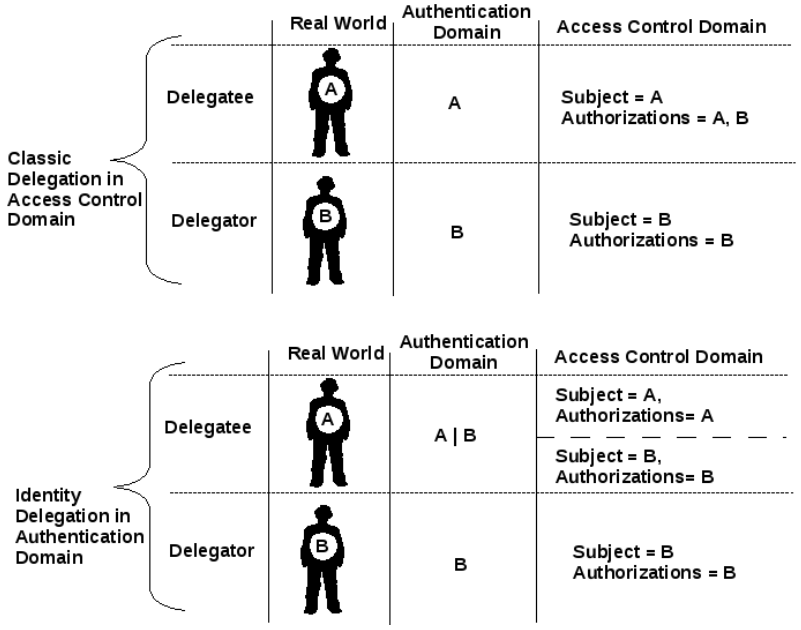


Fig. 2. Difference Between the Two Approaches for Delegation

Figure 2 exemplifies the two approaches for delegation: the delegation in access control domain; and the delegation in authentication domain. In the former case, a validated identity and the corresponding effective identity are assumed to be the same and therefore delegation is managed in the access control domain in terms of permissions and roles; as shown, a delegatee A is recognized as A in the access control domain and has the authorizations of A as well as of B. Delegation in authentication domain is achieved by distinguishing the notions of validated identity and effective identity. As shown in the lower part of the figure, a delegatee A, who is authenticated as A, may assume the effective identity of either A or B. This allows A to use the authorizations of either A or B, depending on the choice of effective identity.

In this paper, we extend the basic definition of identity delegation [2] to include context information. "A context-aware identity delegation at authentication level is a process in which an authentication mechanism provides an

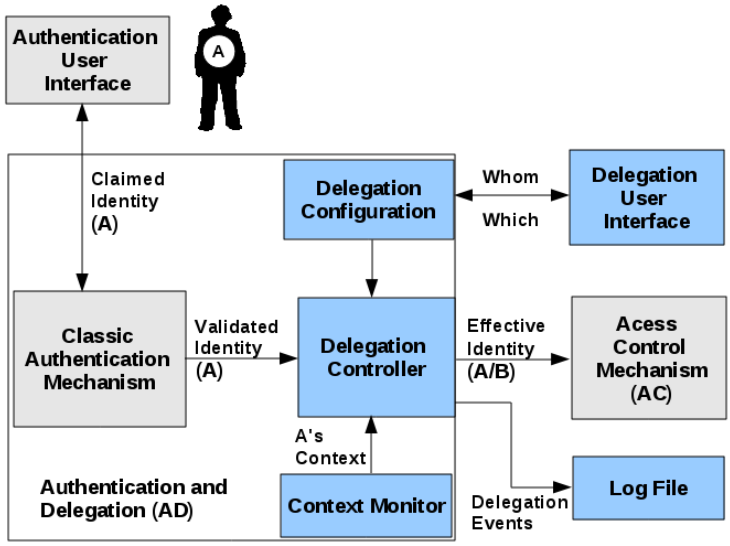


Fig. 3. Context-Aware Identity Delegation at Authentication Level

effective identity that is different from the validated identity of a user provided the following conditions are true."
1. *Whom*: The owner of the effective identity (delegator) has previously delegated his identity to owner of the validated identity (delegatee).
2. *Which*: The current context of authentication for the delegatee is same as previously specified by the delegator.

From the delegator's perspective these two conditions are two simple decisions: *whom* to delegate; under *which* context to delegate. In this paper we only consider time of day and the network address of a computer as the context information, but the definition of context is not limited to these two factors.

The architecture of proposed mechanism is shown in Figure 3. In our architecture, Authentication module is augmented by the three other modules and is shown as Authentication and Delegation(AD) part in Figure 3. These three modules are Delegation Configuration, Delegation Controller and Context Monitor. The Delegation Configuration module contains the delegation policies that are currently active in the system. In the prototype, Delegation Configuration consists of a database, which contains *Whom* and *Which* specifications of the all delegations in a system. The Context Monitor captures and distributes the context information of a delegatee; in our case, it supplies the current time and the network address of the computer on which the delegatee is validated by an Authentication module. The Delegation Controller performs the translation from a validated identity to an effective identity depending on the delegation

policy and input from the Context Monitor; it also records all delegation events in a local log file.

When a delegatee approaches a system, the claimed identity of the delegatee is validated by a classic authentication mechanism, in the usual way. After this, the module Delegation Controller maps the validated identity to an effective user identity based on the input from Delegation Configuration. Now, this effective identity is supplied to the access control mechanism. As defined earlier, this effective identity could either be of the delegatee or of the delegator, depending on the inputs from Delegation Configuration and Context Monitor. The figure shows that a user A can be recognized as a user B in the access control mechanism if B has previously delegated his identity to A and the current context of the system for A is same as previously specified by B.

The configuration in Delegation Configuration can be considered as the delegation policy of a system and is in the form of mappings between validated identity and effective identity. Each mapping relation is associated with some context constraints under which the relation holds. Additionally there is also a list of preferences for each system user that specifies the currently active identity among multiple available identities. Thus, depending on the mapping, context and the preferences a particular identity is provided to the access control mechanism.

We provide a simple user interface in the user's session to specify the delegation policy for new delegations and for changing existing policies. We refer to it as Delegation User Interface in the figure. This interface should only require from a user to decide to *whom* and under *which* context to delegate, so that the user does not worry about specific permissions, roles, security policy or security administrator.

The log file provides a level of accountability in the system. Since our mechanism is at authentication level, we cannot restrict unnecessary delegated authorizations as they are part of the access control domain. This drawback is inherited from the very nature of identity delegation and is justified by the log file and the assumption of mutual trust among co-workers and colleagues [2]. In our mechanism we restrict the propagation of unnecessary authorizations by limiting the delegation in particular context specified by the delegator. In this way we increase the security of a system by limiting the violation of the principle of least privileges.

4 The Prototype

The prototype consists of a personal computer running Debian Linux and is the extended version of our original prototype for identity delegation [2]. We have augmented the classic login based authentication mechanism with RFID based authentication. In the idle state when no user is present on the system, multiple sessions of users are suspended and the computer display is locked. When a user approaches, the relevant session is invoked instantly if RFID badge is validated. This invoked session could be the session of the user or it can be a delegated

session. This selection depends on the preference in the delegation configuration, which can easily be changed by a simple command. The interaction of the

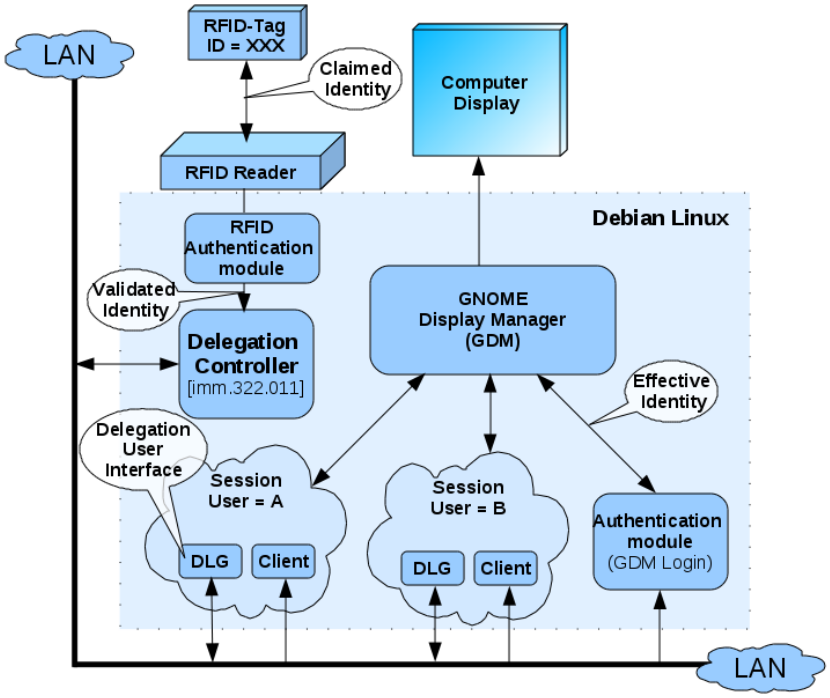


Fig. 4. Architecture of the Prototype

different modules in the prototype is shown in Figure 4. We have not shown Delegation Configuration and Log file. Context Monitor, which in our case consists of the system timer and the network address, is also not shown. RFID Authentication module grants or denies an authentication request using the identification which it receives from the RFID reader. Delegation Controller is launched with the privileges of "root" at start-up, just before GNOME display manager(GDM) is started. In the GNOME based desktop, a single GDM manages the sessions (X-Servers) of all users. The Delegation Controller in the prototype interacts with GDM, by sending commands in a customized format through Clients that run in each of the user's sessions. GDM provides an abstraction for multiple sessions and also invokes the password based login program (GDMlogin). However, GDMlogin program is only a front end for interacting with a user, the actual authentication is achieved by a pluggable authentication module associated with GDM. We use network sockets for the interprocess communication, in order to simplify its porting on distributed nomadic networks.

In typical use, a user carrying a valid RFID tag enters into the interrogation field of an RFID reader. The RFID reader reports the identification to the RFID Authentication module, which checks this identification in the local database to find a match for a system user. If a match is found, the Delegation Controller maps the matched identification (validated identity) to a new identification (effective identity) based on the status of the delegation configuration and the current context. As shown in the figure, the context in our case is limited to a pseudo location *imm.322.11* and the time of day. After deciding the effective identity, Delegation Controller activates the corresponding session.

For the delegation purpose, we have developed a console program in user's space, which represents the user interface for delegation. For example, if a user Bob wants to delegate his session to a user Alice, for the office hours on a computer in Room 11 of Building 322, he issues the following simple command.

```
>dlg set Alice @imm.322.011 [0800-1600]
```

Now, if Alice is in the right context, i.e. Room 11 of Building 322 between 8 am and 4 pm, then she can invoke B's session in one of the following way. Firstly, if she does not have her local account on the machine then the B's session can be invoked automatically when she walks up to the terminal. Secondly, if her local account exists on the machine and she has previously specified her preference for B's session on the machine then again the session can be invoked automatically. Lastly, if her local account exists but she has not previously used B'session then she need to issue following simple command from the terminal.

```
>dlg switch Bob
```

And finally when Bob want to revoke the delegation, he uses the following command.

```
>dlg reset Alice
```

Besides this basic example, there are many more user-friendly commands for different aspects of delegation. The use of location and time in the example is only one possible form of context and in principle, other parameters can be included, though we have not implemented them in the prototype. All the commands of delegation in the prototype share a common motive, which is the simplicity of user interface as we aim to make the delegation process to be more user friendly so that nomadic users actually use it rather than to circumvent it. The syntax of the delegation program is as follow.

```
                Syntax::= dlg <action> <parameters>
              <action>::= set | reset | switch | reset-rec |
                          get | reset-all | NULL
       set:<parameters>::= <user_login_name> <context>
             <context>::= @<location> [<time_period>]
      reset:<parameter>::= <user_login_name>
reset-rec:<parameters>::= NULL
```

```
      get:<parameters> ::= NULL
reset-all:<parameters> ::= NULL
```

Invoking *dlg* without any argument shows help for the command. Otherwise, *set* is for setting an outbound delegation, *reset* is for revoking an existing outbound delegation, *switch* is to switch among delegated sessions, *reset-rec* is to revoke all inbound delegated sessions, *get* is for getting information from existing delegation policy, *reset-all* is for resetting all outbound delegations in one go.

5 Comparison and a Formal Model

Similar to any delegation mechanism developed in the authentication domain, our mechanism violates the principle of least privileges [3], but we believe that this is justified in nomadic environments, where there are pre-established trust relationships among users [2]. The use of context limits the proliferation of unnecessary authorizations, which makes our mechanism more secure than a simple identity delegation at authentication level.

If the delegation is fine grained, then there is always a risk of under-delegation, which effectively results in a denial of service; this is very undesirable in most cases. Similarly, revocation becomes non-trivial in some fine grained delegation mechanisms designed at access control level. In our mechanism, delegation and revocation are just a matter of issuing a simple command.

The classic access control models of UNIX and Unix-like systems may achieve delegation by changing file permissions and group memberships, but this also implies that one should be either the owner of the resource or must have super user privileges; A user is not necessarily able to delegate all privileges which he is authorized to use. In our mechanism this problem is removed and delegation is a user level decision.

In order to compare our mechanism with classic delegation techniques we use a formal logic [18, 1, 12]. We start by introducing a few notions from this logic. When we write **A says S**, it implies that the entity A utters the statement S and believes in its truthfulness. We write $\mathbf{A} \Rightarrow \mathbf{B}$ to represent a kind of trust relationship in a way that whenever A makes a statement, B makes it too. Since the statement, made by a particular entity of the system, implies that the entity believes in it, thus the statement $\mathbf{A} \Rightarrow \mathbf{B}$ represents the trust of B on A. We write **A as R** to represent the entity A in a particular role R and consequently **A as R** has a subset of all authorizations that A normally possesses. For instance, $\mathbf{A'token}$ represents the role of A that possesses the correct authentication token of A. Similarly, $\mathbf{A'context}$ represents a particular context of A. We write (**A** | **B**) **says S** to represent that A is quoting a statement, S, of B. We write (**A for B**) **says S** to represent a quoted statement S and in addition, A is authorized to make such quoted statements on the behalf of B.

Each entity has a set of authorizations, which it can transfer or delegate to other entities in the system using following axioms.

If **A says** (**B** ⇒ **A**) then (**B** ⇒ **A**) for the system ... (1)
(This axiom represents the transfer of authorities from A to B.)

If **A says** ((**B** | **A**) ⇒ (**B for A**))
then ((**B** | **A**) ⇒ (**B for A**)) for the system ... (2)
(This axiom is the classic identity delegation in access control domain as the identity of A is retained)

Now, let us consider Figure 3 and divide it in three system level entities: the user, A; the authentication and the delegation primitives enclosed in the rectangle of the figure, AD; and the access control mechanism, AC. We also assume that the only role of the authentication module AD is to provide the identity of a user. The default trust relationships between these entities are represented by the following statements.

AD ⇒ **AC** ... (3)
(The access control mechanism trusts the authentication mechanism and thus the access control mechanism believes in the identity provided by the authentication mechanism.)

A as A′token ⇒ **AD** ... (4a)
and **B as B′token** ⇒ **AD** ... (4b)
(The authentication mechanism trusts a user with the role of possessing the correct authentication credentials in the form of a valid token that corresponds to the user in the authentication database.)

The statements (3) and (4) are enough to complete the trust chain from a user to the access control mechanism in order to invoke a user's session. For example a user A with A'token can invoke A'session. In our context aware identity delegation mechanism, a user A may delegate his identity to another user B by sending a request to the authentication mechanism in the following form.

(**A as A′token**) **says**
(((**B as B′token**)**and**(**B as B′Context**))|(**A as A′token**))
⇒ **A as A′token**) ... (5)
(An authenticated user A tells the authentication mechanism that if the authenticated user B, in a 'Context', quotes a statement of the authenticated user A then it must be considered as the actual statement from the authenticated user A. The user A specifies Context, in the form of environmental constraints, such as time, location, etc.)

Due to the formal logic axioms, the effect of the statement (5) is the following new trust relationship, which is in fact the context-aware identity delegation.

$$(((\mathbf{B\ as\ B'token}) \mathbf{and} (\mathbf{B\ as\ B'Context})) | (\mathbf{A\ as\ A'token}))$$
$$\Rightarrow \mathbf{A\ as\ A'token} \quad \ldots (6)$$

The statement (6) represents that if a user B, in a specific context, requests for the session of user A, then this request is considered to be equivalent to the request directly made by the user A, which was in form of the statement (4a). Note that the identity delegation represented by (6) is different from the transfer of authorities in (1) and the classic delegation of (2). In (1) the identity of the delegator A is completely lost and in (2) the identity of delegator is present in all requests made by the delegatee. In our mechanism of (6) the identity of the delegator A is only present in the authentication and thus the access control mechanism does not distinguish between a delegator and a delegatee.

6 Conclusions

In nomadic environments, the usability of delegation is a decisive parameter in determining the effective level of system security. People in such environments mostly delegate to their co-workers and colleagues in whom they trust. This fact along with use of context and the logging of delegation events can be used to justify the security of our mechanism in nomadic environments. Since we provide a user friendly interface for delegation, thus many more people will use this secure way of delegation rather than attempt to deceive the access control mechanism for delegating their authorities, which helps to improve the effective level of security.

Despite the obvious security improvement, however, we have not conducted an evaluation to determine exact quantitative values and therefore our claims are only justified by intuitive arguments of the paper; nevertheless, finding good metrics for the accurate measure of security is an open field of research [11].

References

1. Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 15(4):706–734, September 1993.
2. Naveed Ahmed and Christian D. Jensen. A mechanism for identity delegation at authentication level. In *The 14th Nordic Conference in Secure IT Systems, NordSec-2009*, Oslo, Norway, October 2009.
3. Mehran Ahsant. *On-demand Restricted Delegation: A Framework for Dynamic, Context-Aware, Least-Privilege Delegation in Grids*. PhD thesis, Kungliga Tekniska Högskolan, 2009.
4. Girish Chafle Arun Kumar, Neeran Karnik. Context sensitivity in role-based access contro. *ACM SIGOPS Operating Systems Review*, 36(3):53–66, July 2002.
5. Jakob Bardram, Thomas K., and Christina Nielsen. Mobility in health care - reporting on our initial observations and pilot study. Technical Report CfPC 2003-PB-52, Center for Pervasive Computing, 2003.

6. Ezedin S. Barka. *Framework for Role-Based Delegation Models*. PhD thesis, George Mason University, 2002.
7. Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):191–233, August 2001.
8. Michael J. Covington, Srividhya Srinivasan Wende Long, Anind K. Dev., Mustaque Ahamad, and Gregory D. Abowd. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 10–20, Chantilly, Virginia, United States, 2001.
9. M. Gasser and E. McDermott. An architecture for practical delegation a distributed system. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, California, U.S.A., 1990.
10. H.M. Gladney. Access control for large collections. *ACM Transactions on Information Systems*, 15(2):154–194, April 1997.
11. Dieter Gollmann. *Computer Security 2e.* John Wiley and Sons, 2005.
12. Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems (TOCS)*, 10(4):265 – 310, November 1992.
13. Min Li and Hua Wang. ABDM: An extended flexible delegation model in RBAC. In *Proceedings of the 8th IEEE International Conference on Computer and Information Technology*, pages 390–395, Sydney, Australia, July 2008.
14. Dwayne Mercredi and Rod Frey. User login delegation. United States Patent Application Publication, US 2004/0015702 A1, January 2004.
15. J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of IEEE*, 63(9):1278–1308, September 1975.
16. Vijay Varadharajan, Philip Allen, and Stewart Black. An analysis of the proxy problem in distributed systems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, California, U.S.A., 1991.
17. Hua Wang and Jinli Cao. Delegating revocations and authorizations. *Springer Berlin / Heidelberg Lecture Notes in Computer Science(LNCS)*, 4928:294–305, February 2008.
18. Edward Wobber, Martín Abadi, Michael Burrows, and Butler Lampson. Authentication in the taos operating system. *ACM Transactions on Computer Systems (TOCS)*, 12(1):3–32, February 1994.
19. Longhua Zhang, Gail-Joon Ahn, and Bei-Tseng Chu. A rule-based framework for role-based delegation and revocation. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):404–441, August 2003.

POIsafe: a Privacy-Conscious System for Retrieval of Points of Interest

Daniele Riboni, Linda Pareschi, and Claudio Bettini

Università degli Studi di Milano, DICo, Milan, Italy
{riboni,pareschi,bettini}@dico.unimi.it

1 Introduction

Services for retrieval of points of interest (POIs) are becoming increasingly popular due to the widespread diffusion of GPS-enabled mobile devices having access to fast wireless networks. We have developed a context-aware service to share, manage, and retrieve geo-referenced resource descriptions enriched with multimedia content [1]. The access to such services is prone to potentially serious privacy issues, since requests include sensitive information or can lead to the disclosure of sensitive information, and they are often handled by untrusted parties, or sent through insecure channels. Context data, including user location, is in some cases sensitive information that users prefer not to be associated with their identity. In other cases, the interest for specific resources is considered sensitive and the issuer of such a request uses a pseudonym not to be identified; however, context data present in the same request or in a sequence of requests may be used by an adversary to re-identify the issuer. We are not aware of any context-aware service for retrieval of POIs with an effective and comprehensive privacy protection mechanism, and we believe this is a challenging research goal. In this paper, we focus on one particular kind of context data, location, but we plan to extend our techniques to tackle the general problem illustrated above.

Different techniques have been proposed for protecting against the disclosure of location information in location-based services (LBS). Cryptographic approaches inspired by Private Information Retrieval (e.g., [2]) provide very strong guarantees in terms of privacy; however, they determine a relevant overhead in network and power consumption and service response time, especially when applied to services that consider a wide set of context data. Obfuscation-based techniques are based on a perturbation of the user's location. In techniques based on generalization the exact location is enlarged to a region; in other cases, a fake user's location is communicated instead of the real one. In particular, the latter approach is adopted by *SpaceTwist* [5] to enforce location privacy while guaranteeing that k-nearest neighbor (kNN) queries are correctly answered even if the user provides a fake location, at the cost of computation and communication overhead. Indeed, according to SpaceTwist, the client issues a sequence of requests from the same fake location asking for more close-by POIs until it is sure that those provided by the service include the kNN set corresponding to its real location. Based on the request-response sequence, an adversary can

only identify an area (called *twisted space*) from which the requests may have been sent. More recently, a technique (derived from SpaceTwist) to couple location privacy with identity anonymity, named *AnonTwist* [4], has been proposed. Given a density map to estimate the number of people in a given area, AnonTwist provides a probabilistic guarantee that, even if an adversary has access to presence information, the twisted space contains at least N individuals; hence, the request issuer is indistinguishable among at least N individuals. However, both SpaceTwist and AnonTwist rely on the assumption that the function that generates the fake location is unknown to the adversary.

In this paper we take into account the realistic case in which the function that generates the fake location is known to the adversary. In Section 2 we illustrate a specific technique to protect privacy under this assumption. In Section 3 we present *POIsafe*, an extension of our system for POIs retrieval to enforce location privacy and identity anonymity as a first step towards a comprehensive solution considering other context data. Section 4 concludes the paper.

2 Enforcing privacy in *POIsafe*

Even if we are investigating alternative solutions, our current approach is inspired by AnonTwist [4]. However, differently from AnonTwist and SpaceTwist, our proposed algorithm assumes that the function for generating fake locations could be known to the adversary. Under this assumption, by observing the fake location, the adversary may reconstruct the possible area A from which it originated. Moreover, based on the request-response sequence, the adversary is able to understand that the area from which the request originated corresponds to the intersection I between the twisted space and area A. Hence, the goal of our technique is to ensure that area I is greater than a specified threshold, and that it includes at least N potential issuers.

In particular, the maximum radius r of perturbation (e.g., 1 Km) and the minimum number of potential issuers N are chosen according to the user's preferences about privacy and quality of service (QoS). Then, before issuing a request, a random *distance* ranging from 0 to r is chosen, and a random point having that distance from the real user's location is chosen as the fake location. Note that with this technique, A is the area contained within the circle having center in the fake location, and radius r. Then, the client incrementally asks for nearest POIs until *i)* the exact kNN set is retrieved, *ii)* area I is greater than the chosen threshold, and *iii)* I contains at least N users according to the density map.

In general, the farther is the fake location from the real one, the higher is the user's privacy. However, large perturbations of the real location determine poor QoS. Indeed, even if the service guarantees to correctly answer the user's query, a very high number of POIs may be communicated to the client before obtaining the correct kNN set, determining an increase in communication and computational costs, and response time. With respect to usability, users can be provided with an intuitive interface to set their preferences; i.e., a single slider with *response time* on the left-hand side and *privacy* on the right-hand side.

The value of the slider influences the value of radius r and threshold N. Before submitting a request, density information is used to control if the value of r is adequate to provide a sufficient level of anonymity with high probability (e.g., according to the density map the corresponding area includes a number of users that doubles threshold N). In this case, a green signal appears; in the other case, a red one is used. Note that the red signal does not mean that the technique will necessarily fail to preserve privacy, but only that anonymity might be at risk.

3 System architecture

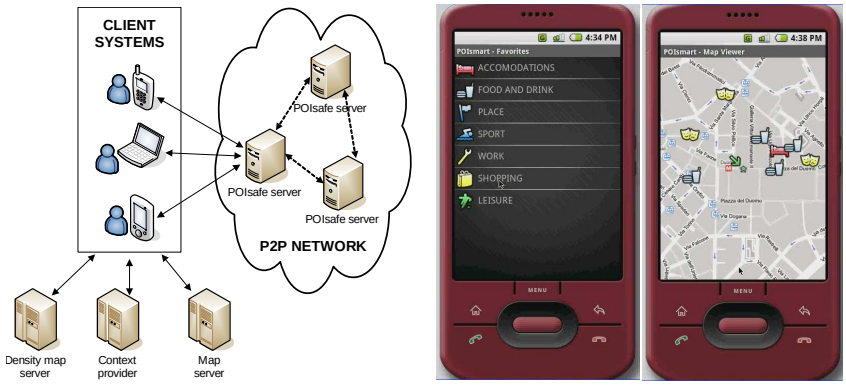
POIsafe is based on a peer-to-peer network of POISAFE SERVERS, which are in charge of managing POIs, and of searching POIs in the peer-to-peer network on the basis of the user's context and explicit search keywords. The mechanism of search in the peer-to-peer network has been presented in detail in [1]. An overview of the *POIsafe* network is shown in Figure 1(a).

Users can access the *POIsafe* network from a wide range of CLIENT SYSTEMS, which provide an interface for the user to browse her own POIs hierarchy, reorganize the hierarchy, add new POIs, search shared POIs in the peer-to-peer network, and set their preferences, including those regarding privacy. Before issuing a request, each CLIENT SYSTEM retrieves context information useful for service adaptation. This information can be retrieved either locally or from an external CONTEXT PROVIDER. Perturbation of location information and requests for POIs are executed as illustrated in Section 2. The density map is retrieved from a trusted density map server. In the previous version of *POIsafe*, ranking of POIs was performed only at the server side. However, since in the new version a fake user's location is communicated to the server, returned POIs are re-ranked at the client side considering the exact user's location. Moreover, an external MAP SERVER is queried by the CLIENT SYSTEM for obtaining maps showing the position of returned POIs and information for navigation support.

The POISAFE SERVER has been developed in Java, and implements the algorithms for POIs scoring and distributed search presented in [1], the modified AnonTwist algorithm, as well as various facilities for managing and searching POIs. The architecture adopts Web services for client/server and server/server communication. At the time of writing we have developed CLIENT SYSTEMS for laptops and smartphones. In particular, we have developed a novel Android client (see Figure 1(b)), which takes advantage of the integration with Google Maps.

4 Conclusions and future work

In this paper we presented the extension of an existing system for context-aware management and retrieval of points of interest. The extension is a first step towards a comprehensive privacy solution for this kind of services. In particular, future work includes a thorough investigation of the formal properties of the proposed algorithm. Several research issues remain open. In particular, we point out that the proposed technique may be ineffective if an adversary can observe



(a) *POIsafe* network (b) Client implementation

Fig. 1. *POIsafe* system and current implementation

histories of requests issued by users in different time granules. Indeed, as shown in [3], the frequency of a service parameter included in requests, matched with the presence of candidate issuers, can be exploited to associate a given user with that service parameter. The integration of techniques to protect against these kinds of attacks will be the subject of future investigation.

Acknowledgments

The authors would like to thank Song Wang and X. Sean Wang for providing a working implementation of the AnonTwist algorithm.

References

1. C. Bettini and D. Riboni. Context-aware Web Services for Distributed Retrieval of Points of Interest. In *Proc. of the 2nd International Conference on Internet and Web Applications and Services*, page 36–40. IEEE Computer Society, 2007.
2. G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private Queries in Location Based Services: Anonymizers are Not Necessary. In *Proc. of SIGMOD 2008*, pages 121–132. ACM, 2008.
3. D. Riboni, L. Pareschi, C. Bettini, and S. Jajodia. Preserving Anonymity of Recurrent Location-based Queries. In *Proc. of the 16th International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society, 2009.
4. S. Wang and X.S. Wang. AnonTwist: Nearest Neighbor Querying with Both Location Privacy and k-Anonymity for Mobile Users. In *Proc. of First International Workshop on Mobile Urban Sensing*, pages 443–448. IEEE Computer Society, 2009.
5. M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu. SpaceTwist: Managing the Trade-offs among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In *Proc. of ICDE 2008*, pages 366–375. IEEE Computer Society, 2008.

Portable reputation: Proving ownership of reputations across portals

Sandeep S. Kumar and Paul Koster

Information and System Security Group
Philips Research Laboratories, Eindhoven, The Netherlands
{Sandeep.Kumar,R.P.Koster}@philips.com

Abstract. User reputation has become a valuable commodity for enabling trusted transactions on the Internet especially with strangers in virtual communities. However, the reputation information about the various users are normally locked-up in the silos of different web-portals where the members interact. This effectively creates multiple reputation ratings for the same user, each one painstakingly built over a long time at each web-portal. Though this status quo is favorable for established web-portals as it enables them to lock-in their customers, consumers have a strong interest in a portable reputation system that allows them to cross the boundaries of the competing portals. In this paper, we present a portable reputation mechanism which is managed by the users on their own with minimal co-operation from the web-portals. This method enables a user to combine from various portals the reputation information of others, which are proven to belong to them in a reliable and cryptographically secure way. Users can appropriately weigh the reputation from different web-portals according to their individual choice and trust in the different web-portals. The solution has the advantage that it does not require any unified reputation rating framework implemented by all web-portals. Additionally the cryptographic binding is constructed such that it prevents users to form a coalition and share their good reputation ratings among them.

1 Introduction

In today's Internet connected society, it has become more common to have new forms of interactions with complete strangers. These virtual interactions can be for various purposes like getting the right information (like Yahoo answers), for real value transactions (like Ebay), and many more. All such interactions with strangers require some amount of trust in the other party which goes beyond the knowledge of their virtual pseudonymous identity. In fact, to encourage these forms of stranger-stranger interactions, web-portals use reputation based systems which are normally some form of numerical ratings of the past behavior. The reputation system functions by collecting, aggregating and distributing the historic behavior of the participating entities of the web-portal. Resnick et. al. [6] mention three properties that a reputation system requires at a minimum:

1. Participating entities are long-lived, so that there is an expectation of future interaction.
2. Feedback about current interactions between entities are captured and distributed. Such information must be visible in the future.
3. Past feedback should guide new interaction decisions. Therefore entities must pay attention to the reputation while making their decisions.

Ebay is an example where the reputation system is attributed to its phenomenal success in enabling real valued auctions between complete strangers [2]. The reputation information in such systems gives the parties involved in the interaction the much needed help in deciding whom to trust, encouraging trustworthy behavior to maintain high reputation, and deter participation of cheater and unskilled parties. Reputation therefore has become an extremely valuable commodity which enables higher price premiums in identical transactions [7] and people work hard to earn reputation on these portals. Consequently for web-portals, reputation not only enables new transaction models and possibilities but also provides the ability to lock-in customers. Customers who have worked hard on creating a good reputation on a portal are less likely to switch to another portal unless the benefit is substantially more than the effort required to recreate a similar good reputation. However, a user switching to a new portal not only needs to painstakingly rebuild his reputation but also has to lose out on his higher price premium which he would have been entitled based on his higher perceived trustworthiness. These silos of reputation can follow from a combination the reasons listed below.

1. Different portals cannot securely verify if the pseudonyms used on both portals actually belong to the same user. This can be partially solved if a federated identity management system is in place. However, this does not scale well with the large number of portals which need to be within the *circle of trust* of the federated identity provider.
2. Portal owners realize that the painstakingly generated reputation ratings enable them to lock-in users to their portal and hence would not easily participate in a federated system in which they need to share detailed information about their users with possible competitors.
3. Different portals use completely different reputation rating frameworks which do not directly map to each other. The different frameworks are used because each web-portal considers different aspects as important to rate trustworthy behavior. Other reason can also be due to the wide variety of ways that the reputation systems can be attacked [4] and by making a system closed and controllable, some of the attacks can be thwarted.

However, portability of reputation data can be of utmost importance to consumers. Due to the reputation lockup, consumers are implicity forced to use their current portals for their interactions even if they find other new portals more attractive. An alternative for this is to have a centralized server which handles all the reputation data of users across portals (a similar idea is mentioned in [6], tried by virtualfeedback.com and now defunct). The problem with a

centralized approach is that different systems require different reputation frameworks as mentioned before. The OASIS Open Reputation Management Systems (ORMS) Technical Committee [1] is trying to address standardization and interoperability of information used to derive reputation ratings for individuals and institutions participating in Internet communities. However there are other reasons too which favor a controlled closed system, for example, it is hard to make sure that the reputation ratings are being provided by actual transaction partners, which can be reliably verified only by the web-portal where the transaction had occurred.

1.1 Our Contribution

In this paper, we map the problem of portability of reputation information to users wanting to show other entities their reputation ratings on different web-portals and prove that those belong to him/her. It is normal for the reputation information of an user of a web-portal to be publicly visible to everyone. It associates the reputation information to the user's pseudonym on the portal. Therefore proving possession of reputation can be addressed by proving possession of the pseudonym at the portal. The entity to whom the reputation ratings are shown can decide how to weigh those ratings based on their trust in the originating portal or their trust metric (based on context of the present interaction).

The method described in the paper enables users to easily claim their local reputation at various web-portals by provably presenting the possession of pseudonyms to a requesting entity to increase the level of trust in an interaction. The requesting entity could be a new transaction partner or a web-portal allowing users to initially use the reputation data from other web-portals to reduce the burden of starting off as an unknown entity.

The paper is organized as follows: in Section 2 we specify the problem and assumptions of the system. We outline the solution in Section 3 and in Section 4 we present some cryptographic preliminaries and the protocol details. Some applications are presented in Section 5 and conclusions are presented in Section 6.

2 Problem definition

Users normally have different pseudonyms at different web-portals where they engage in transactions and earn reputation. For example, an user *Alice* can have pseudonyms $P1$, $P2$ and $P3$ on three different portals $S1$, $S2$ and $S3$ respectively as shown in Fig. 1. These pseudonyms can be directly linked to a real user if personally identifiable information is revealed (like Alice's name or email-address) along with the pseudonyms. However, it is also very easy for an impersonator to claim to be Alice on a different portal by mentioning the corresponding personally identifiable information. Hence the link-ability is not strong unless such information is cryptographically signed. Furthermore, making such personal information public leads to the problem of privacy. Most users tend to have non-identifiable pseudonyms on different portals for privacy reasons to avoid linking

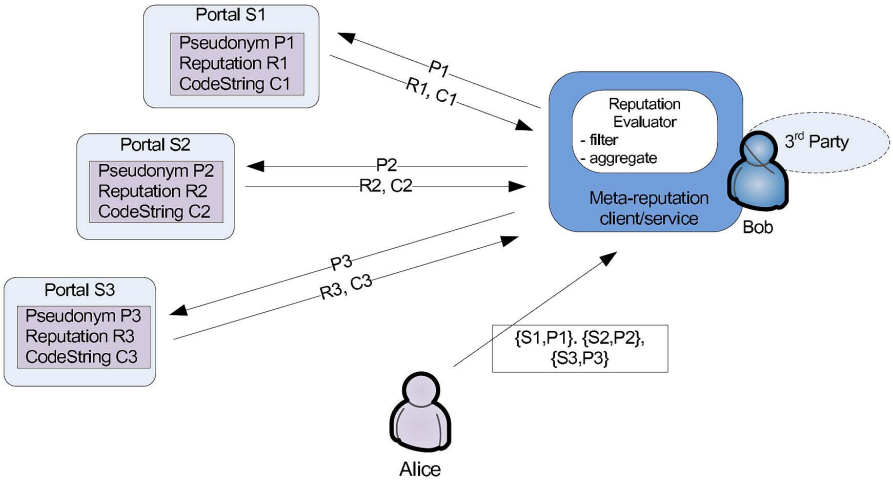


Fig. 1. Architecture for portable reputation

of profiles and their actions on these portals. Cryptographic signatures can only further help in this linking process and consequently reduce privacy. Anonymous linking of pseudonyms is possible if a trusted identity provider is acting as the pseudonym provider (or identity provider) for all the portals as is the case in federated identity management. However, most of the time services are competitors who value their customer base and do not trust each other. This leads to the need for a solution where the user Alice would like to reveal possession of the pseudonyms $P1$, $P2$ and $P3$ to a 3rd party like *Bob* without the active involvement of the web-portals $S1$, $S2$ and $S3$. This enables Alice to claim her reputation $R1$, $R2$ and $R3$ at the various portals and reliably prove to Bob as belonging to her. Bob can then decide how to combine these reputation ratings from different portals based on his trust in how reputation is evaluated at these portals or based on the nature of the current interaction with Alice.

3 Solution Outline

The main goal of the solution is to allow users to make their reputation portable on their own with minimal interaction between portals. The first step in the process is for the user (here Alice) to create some form of a coded string and attach it to her pseudonym in a publicly accessible way (e.g. the user profile page of Alice as shown in Fig. 1 as $C1$, $C2$ and $C3$). The purpose of this coded string is to enable Alice to prove that she possesses the corresponding pseudonyms and therefore the associated reputation. This coded string needs to satisfy the following requirements:

1. The coded string contains a secret that only the user (Alice) is aware of and can prove knowledge of it (preferably without revealing the secret)
2. The coded string should be bound to the user (Alice) in such a way that she cannot transfer her reputation to other users without revealing a secret that compromises all of her other pseudonyms
3. The coded string should not be a source of information to enable linking of pseudonyms by third parties not involved in the transaction with the user (Alice).

To help Bob verify the coded string, we define a meta-reputation (meta-RS) client to which Alice can provably claim all the pseudonyms as belonging to her. The meta-RS also enables the aggregation of the reputation ratings based on Bob's preferences. An alternative is to consider the meta-RS to be a trusted third party which both parties trust to perform truthfully all the various verification operations. The meta-RS as an intermediate trusted party is suitable in environments where Alice does not want to reveal her pseudonyms to Bob but only wants to give Bob the possibility to get an aggregate reputation value based on his preferences. The other advantage of a trusted intermediary is that no client is required at the transacting party Bob and instead could be a third party web-site which performs all the operations of the meta-RS client.

4 Detailed Protocol Description

Before we describe the solution, we first present the relevant cryptographic concepts that constitute the solution.

4.1 Cryptographic preliminaries

The main underlying concept is the computational hardness of the discrete logarithm (DL) problem which is defined as follows

Definition 1 *Given a finite group* $\mathbf{G}$*,* g *the generator element, and* $e \in \mathbf{G}^*$*, find* m *such that* $g^m = e$*.*

The DL problem exists in any group, however when used for cryptographic purposes the group is usually chosen as the multiplicative group of integers modulo N, $\mathbb{Z}_N^*$, where N is chosen to be a relatively large prime. Various well established public-key cryptosystems are based on the DL problem like the ElGamal [3] system and the Digital Signature Standard (DSS) [5].

Based on the DL problem is the equally hard Diffie-Hellman (DH) problem which is defined as

Definition 2 *Given a finite group* $\mathbf{G}$*,* g *the generator element, and randomly chosen* $x, y \in \mathbb{Z}$*. Given* g^x *and* g^y*, find* g^{xy} *.*

4.2 Pseudonym ownership protocol

We present here one method to generate the coded string that can fulfil the requirements mentioned before. For the representation of the coded string and the protocol, we use the following notation

1. N is a prime and all arithmetic is performed in the multiplicative group of integers modulo N. N is chosen relatively large to be secure based on the hardness of the discrete logarithm problem. N is publicly known.
2. g is a generator of the multiplicative group modulo N. g is publicly known.
3. $H[\,]$ is a hash function, e.g. SHA-256
4. I is the user and U_I is the unique secret known only to him.
5. P_i are the pseudonyms associated with user I at web-portals S_i
6. K_i are the unique secrets associated with each pseudonym P_i
7. a and b are random numbers

The coded string used by user I (say Alice) for the pseudonym P_i is the pair constructed as follows $\mathbb{C}_i = (\bar{c}, \hat{c}) = (g^{K_i}, g^{U_I K_i})$.

Alice wants to prove to a third party (say Bob, who could also be a web-portal) that a set of pseudonyms at web-portals $\mathbb{P} = \{\{P_1, S_1\}, \ldots, \{P_n, S_n\}\}$ belongs to her and therefore the corresponding reputation which is publicly visible at these portals. The protocol is shown in detail in Fig. 4.2 and each of the steps is described further.

Bob chooses a random number a which he keeps secret to himself and sends the value $\alpha = g^a$ to Alice. Alice cannot determine a from α based on the discrete log problem. However Alice can use her secret K_i to derive $\alpha^{K_i} = g^{a*K_i}$. Bob similarly has the value $\bar{c} = g^{K_i}$ but not the value K_i. Bob performs a similar step $\bar{c}_i^a = g^{K_i * a}$. Hence now Alice and Bob have the same value. An additional hashing is done by Alice to derive $\nu_i = H[\alpha^{K_i}]$ and sends it to Bob. If Bob can verify that ν_i is same as the hash value he generates $\tau_i = H[\bar{c}_i^a]$, then it is clear to Bob that Alice knows the secret and has access to the pseudonym. This step is performed for each of the pseudonyms that Alice shows to Bob.

However, the fact that Alice knows the secret for each individual pseudonym does not prohibit Alice to form a coalition with multiple people and use a secret of an account belonging to a member of the coalition for this particular interaction. Therefore Bob needs to be convinced that all accounts have something in common (here the secret U_I) that is known only to Alice.

Now Bob chooses a random number b which he keeps as secret to himself and computes

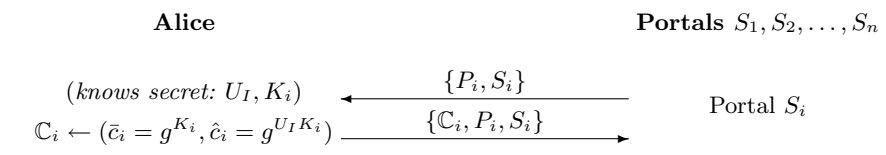
$$\begin{aligned}\omega &= (\bar{c}_1 * \bar{c}_2 * \ldots * \bar{c}_n)^b \\ &= (g^{K_1} * g^{K_2} * \ldots * g^{K_n})^b \\ &= g^{(K_1 * K_2 * \ldots * K_n) * b}\end{aligned}$$

and sends ω to Alice. Alice now computes

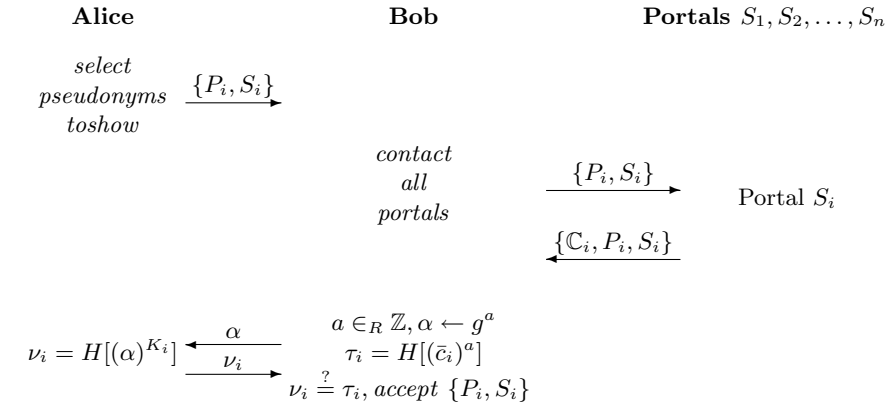
$$\begin{aligned}\varphi &= H[\omega^{U_I}] \\ &= H[(g^{(K_1 * K_2 * \ldots * K_n) * b})^{U_I}] \\ &= H[(g^{(K_1 * K_2 * \ldots * K_n) * b * U_I}]\end{aligned}$$

1. **Assumptions:**
   - Communication channel between all parties are authenticated
   - N and g public

2. **Creating code-string for pseudonym ownership:**



3. **Proving ownership of individual pseudonyms:**



4. **Proving all pseudonyms have the same owner:**

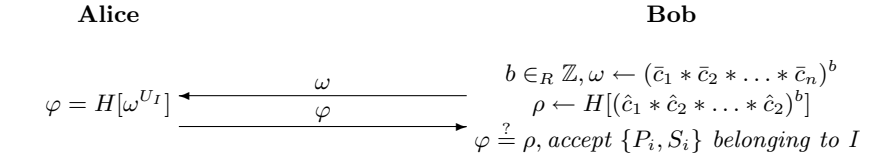


Fig. 2. Pseudonym ownership protocol

and sends φ to Bob.

Bob can check if all pseudonyms belong to Alice by first computing

$$\begin{aligned}\rho &= H[(\hat{c}_1 * \hat{c}_2 * \ldots * \hat{c}_2)^b] \\ &= H[(g^{U_I K_1} * g^{U_I K_2} * \ldots * g^{U_I K_n})^b] \\ &= H[(g^{(K_1 * K_2 * \ldots * K_n) * U_I * b}]\end{aligned}$$

If $\varphi = \rho$, it proves that Alice possesses the secret U_I and hence proves that all the pseudonyms belong to her. The main reason that prevents sharing of

pseudonyms across users is that once a pseudonym is shared that Alice also has to reveal her unique secret U_I, which compromises the security of all her other pseudonyms.

Once Bob is convinced about the ownership of the pseudonyms by Alice, he can contact the portals to retrieve the publicly visible reputation ratings of each of the pseudonyms. Bob's meta-RS can combine these ratings based on the context in which he is interacting with Alice and on his other personal preferences like weights for different portals.

However, it is important to realize certain limitations in the methodology, like users not revealing their pseudonyms with bad reputation. Our intention is not to capture all possible reputation of the user but to give users the ability to prove possession of any reputation they claim to be theirs and leave the the final decision to the users who will need to weigh it for their transactions.

5 Applications

Possible applications of the presented portable reputation mechanism include online services and emerging pervasive applications. Today, rating and reputation-based online services include online auctions, discussion forums, questions-and-answers boards, and review services. These services have in common that users build up a reputation by participating in transactions. Using the portable reputation mechanism enables users to leverage reputation for a broader range of online services, illustrating a typical case of user-centric identity. For example, it would enable somebody to incidentally buy or sell something on a niche site while also presenting his reputation from e.g. E-bay. For this purpose the user subscribes to an independent reputation aggregation service (meta-RS) trusted by both parties. This ensures that he can use his standard browser and avoids the need for a client-side extension. The user establishes an account at this meta-RS service and indicates at which services he holds a reputation. The meta-RS calculates the coded string and instructs the user to include them in his profile at the individual services. Subsequently, a transacting entity and the meta-RS can perform the ownership protocol, enabling the user to bootstrap his reputation in a trustworthy way. The meta-RS service can also present an aggregated reputation omitting the details of the full protocol and hence preserving privacy. It then presents the transacting party with the most relevant and useful information, which may involve selection, annotation and post-processing of reputation according to certain rules to form an aggregate value.

Pervasive applications like monitoring and coaching or independent living are a second class of applications that benefit from a portable reputation mechanism. The combination of sensor-based monitoring and interactions in the physical environment as well as a rich set of third party services makes trust very important. However, these applications tend to build up reputation less quickly since the transaction frequency is lower and distributed over different contexts. Portable reputation would enable a user to leverage his longstanding reputation at a variety of review services and discussion forums for e.g. a peer-coaching

service using reputation-based trust between community members. In this class of applications reputation portability may be facilitated by the user's personal device having a meta-RS client in addition to its functions to aggregate sensor information and interact with services and devices in the physical environment. The personal device and the services can execute the portable reputation protocol. The protocol may also be executed in a peer-to-peer model between users' devices.

6 Conclusions

We presented a method through which users can prove ownership of their reputation across web-portals. We assumed a model of minimal co-operation of web-portals and created a user-centric protocol. The portable reputation mechanism is managed by the users on their own with minimal co-operation from the web-portals. Users can prove to each other (or other parties) in a reliable and cryptographically secure way the possession of the various individual reputation ratings without requiring any unified reputation rating framework implemented by all web-portals. Additionally the cryptographic binding is constructed such that it prevents users from sharing their reputation ratings with others. We also present some application scenarios where it could be used.

References

1. *OASIS Open Reputation Management Systems (ORMS) Technical Committee.*
2. Jennifer Brown and John Morgan. Reputation in online markets: The market for trust. *California Management Review*, 49(1):61–81, 2006.
3. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
4. ENISA. Reputation-based systems: a security analysis. Technical Report 2, ENISA Position Paper, 2007.
5. CORPORATE NIST. The digital signature standard. *Commun. ACM*, 35(7):36–40, 1992.
6. Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.
7. Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The value of reputation on ebay: A controlled experiment. *Experimental Economics*, 9(2):79–101, June 2006.

Identity Metasystem in Location Based Persistent Authentication

Hasan Ibne Akram[1] and Christian Damsgaard Jensen[2] and Mario Hoffmann[1]

[1] Fraunhofer Institute for Secure Information Technology
Munich, Germany
{*hasan.akram,mario.hoffmann*}*@sit.fraunhofer.de*
[2] Technical University of Denmark
Kgs. Lyngby, Denmark
christian.jensen@imm.dtu.dk

Abstract. Persistent authentication in smart environments (PAISE) is a new effective approach that has changed the device-centric paradigm in traditional token based authentication systems to a much more user-centric one. Instead of carrying a security token, such as an RFID tag, all the time the location of a previously authenticated user is simply persistently tracked by sensors in the smart environment, such as cameras. For closed scenarios, recent publications have already shown various advantages, such as consistent user experience across contexts and a decreased probability of impersonation. In order to address even more complex and privacy-enhanced scenarios, we propose to enhance PAISE with Identity Management (IdM). Taking advantage of IdM, users can easily choose among different virtual identities and, at the same time, we can now support open and cross-context application scenarios. This paper addresses the balance between the security requirements of service providers in smart environments and the privacy concerns of users. Therefore, this paper discusses first security and privacy implications in PAISE and evaluates then the adaptability and interoperability of existing identity management systems for persistent authentication in changing and open scenarios. As our main result and in order to ensure interoperability an Identity Metasystem on top of the PAISE architecture will be described in detail.

1 Introduction

Authentication in context aware smart environments is typically device-centric and based on for example RFID tokens, Smartcards or - more advanced - cell or smart phones. Users have to carry these authentication tokens all the time and present them to technical or human safeguards. This makes it relatively easy, however, to impersonate authenticated users by cloning, borrowing or stealing their authentication token. Therefore recent publications [1] proposed a new user-centric paradigm based on user tracking called PAISE - Persistent Authentication in Smart Environments. That means that once you have been authenticated at a particular security gate you will be continuously tracked in a smart

environment, e.g. by sensors or cameras, until you leave the area covered. In this paper we will deal with the implications of this physical single-sign-on experience balancing security requirements and privacy concerns and propose appropriate enhancements to the existing PAISE architecture.

1.1 Persistent Authentication

Persistent Authentication [1] has been proposed as a proactive authentication mechanism, which combines existing authentication mechanisms based on passwords, smartcards or biometrics, with person tracking using "Time-of-Flight" (TOF) cameras. Users are authenticated when they enter the smart environment and the TOF cameras are subsequently used to track the user as he moves around in the smart environment, which makes the event of authentication "stick" to the user, thus making it persistent. This means that the persistent authentication mechanism can proactively authenticate the user to a service in the smart environment whenever the user requests access to that service. Ideally, this means that users would only have to authenticate when they arrive at work in the morning and the system will be able to authenticate them to physical access control services, computing services and other services throughout the day. Similar ideas have been explored in other projects, but these have all required users to carry a small authentication token with them at all times and it is effectively this token that is authenticated, so we refer to such mechanisms as device-centric authentication. The problem with device-centric authentication is that users must always remember to carry the token with them and authentication tokens can be borrowed, lost or stolen. We therefore believe that tracking the user, rather than the token, is a more convenient and secure way to authenticate users in emerging smart environments. However, constant tracking of users will be seen as a serious violation of privacy by many people, so it is important to develop a system that limits the problem of privacy violation in location-based persistent authentication.

1.2 Identity Management

Identity Management (IdM) refers to the maintenance of the complete lifecycle of digital virtual identities. Virtual identities are characterized by a collection of personalized information containing for example names, address(es), eMail and telephone, account data, bank or credit card information, as well as preferences, profiles, histories of service usages and contexts. We can say that virtual identities reflect at least parts of one's real identity and basically consist of an open and dynamic set of attributes for a specific period of time. In smart environments IdM gains even more attention and importance. Smart environments are enabled by IT systems working in the background gathering and reasoning on context information including analyzing and predicting user behavior. On the one hand these systems will provide useful and convenient personalized services in the future - persistent authentication based on TOF cameras is such a service; on the other hand the information collected might - in principle - violate

the users' privacy if you do not establish reasonable mechanisms to tackle these concerns by managing personalized information in a secure way. Identity Management Systems enable both users as well as service providers to manage virtual identities in a secure and effective way. Users are able to maintain for example a set of virtual identities depending on account and authorization information for specific services and contexts. Service providers are at the same time able to distinguish different users acting in different contexts attaching different authorization and access rights in order to assure accountability and the establishment of appropriate security policies. Therefore, we introduce in this paper IdM concepts, including an Identity Metasystem, to PAISE in order to address raising privacy concerns in smart environments equipped with TOF cameras and to balance these concerns with security requirements from service providers. Users who work in such environments IdM helps to choose the right virtual identity with the minimum set of necessary attributes. Service providers who run smart environments relying on persistent authentication IdM helps to attach the right privileges and access rights to users. The proposed Identity Metasystem will finally ensure that different IdM Systems will be able to interoperate across different contexts and scenarios.

1.3 Paper Structure

The paper is organized in the following way: Section 2 presents a short overview of the PAISE model and identifies the privacy concerns that must be addressed if persistent authentication is to be widely deployed. Section 3 introduces the interoperability concept of Identity Metasystems followed by Section 4 where according to the privacy concerns the state of the art in identity management technologies will be analyzed. Section 5 proposes an architecture that integrates Identity Metasystems with the existing persistent authentication architecture and Section 6 presents an evaluation and discussion of the proposed architecture. Finally, we present our conclusions in Section 7.

2 Location Tracking

2.1 The PAISE Model

The PAISE model defines four major components in a persistent authentication system: an authentication system, which is able to authenticate principals; a smart environment, which delivers the sensor data needed for tracking; an access control mechanism, which acts on the result of persistent authentication and the core component of PAISE, which combines the information from the authentication system and the smart environment, tracks authenticated principals in the smart environment and forward the necessary data to the access control mechanism. These components are shown in Figure 1.

In addition to these four components, PAISE also defines authentication zones and authorization zones in the smart environment. An authentication zone

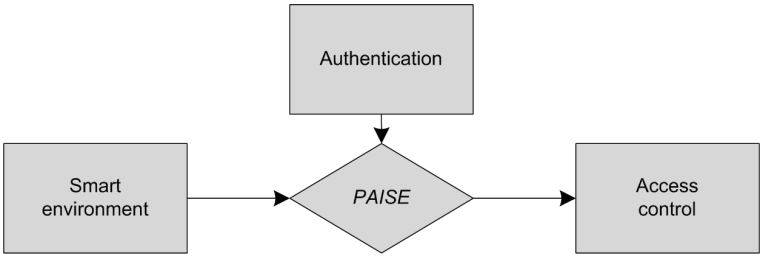


Fig. 1. PAISE Authentication Model.

defines the area in front of the authentication mechanism which is large enough to hold a single principal.

The smart environment delivers a constant stream of sensor data to the core component, but tracking is only initiated when a principal has entered the authentication zone and successfully authenticated himself. The authentication zone must be small enough to ensure that the authentication event can be reliably linked to the principal. A typical authentication zone, in a smart environment, would be an area of 0.5m x 0.5m in front of a swipe-card terminal. An authorization zone defines the area in which the access control policy of a location based service must be enforced. When new principals enter an authorization zone the persistent authentication is forwarded to the access control mechanism of the location based service provider, which is then able to determine whether access should be granted. In the case of access through a door, in a smart environment, the authorization zone must be small enough to ensure that most principals are able to reach and open the door while it is unlocked, but also large enough to ensure that nobody outside the authorization zone is able to pass through the door while it is open. This allows the system to enforce the constraint that the door can only be unlocked if there are no unauthenticated or unauthorized principals inside the authorization zone, thus preventing tailgating.

2.2 Privacy Concerns in Location Tracking Systems in Smart Environments

Privacy issue is by no means a new concern in Ubiquitous Computing or smart environments. The foreseer of ubiquitous computing Mark Weiser had already pointed out the issue of privacy in 1991 [2]. Langheinrich showed how potentially privacy can be endangered in such environments without even the consent of the user [3]. Similarly, location based tracking systems (LTS) has been shown to be inherently privacy dissenting. Therefore, the privacy issue in smart environments using LTS must be taken into consideration and requirements for privacy compliance has to be defined. Previous studies show that privacy risk is apparent on LTSs (e.g. RFID, GPS etc.) since LTSs collect information silently, without the consent or even awareness of the user [4–6]. Information can be collected

about an individual and can be aggregated to figure out near-perfect knowledge of the individual's whereabouts and activities. If we refer back to the definition of privacy from Louis Brandeis, 1890 (Harvard Law Review): "The right to be let alone.", LTSs violate privacy of an individual. Although PAISE is inherently designed to be relatively more privacy compliant than other camera based tracking as it uses a non recognizable image (blob) of the object, still it suffers from some traditional privacy issues that any location based tracking system has. If we get back to the definition of Brandeis and try to apply it in PAISE we see that PAISE is even less compliant to privacy because: in traditional device based tracking system, the user is able to switch off the device when he wants to and thus becomes invisible to the system. On the other hand, it is not possible to switch off tracking in PAISE that easily, so the user's right to be left alone is not easily accomplished.

2.3 Privacy Principles in Smart Environments

Requirements and principles of Identity Management have been analyzed and derived in pervasive computing ever since the very beginning of pervasive computing. Obviously, these related works have some commonalities and disparities among themselves. Our objective in this section is to narrow down the privacy principles of Smart Environments suitable to LTS in such environments. Langheinrich [3] identified six principles and guidelines for privacy aware ubiquitous system. It is important to mention that these principles and guidelines do not ensure total privacy. The goals of these privacy guidelines are to get as close as possible to Brandeis' definition of privacy. We have picked four of the privacy principles as they are mostly relevant for persistent authentication using LTS. These four principles are outlined below:

- **Notice:** "Transparency" or "Openness" is the most fundamental principle of any data collection system. When the location data about the user is collected in PAISE but not saved, the user should be notified about that.
- **Choice and Consent:** A mere notice to the user about its location data being tracked is not enough anymore as EU Directive [3] refined and extended the well-known fair information practices. There has to be explicit consent of the user about the location tracking.
- **Anonymity and Pseudonymity:** Since in a LTS it is very difficult to have an explicit consent of the user at any given time, a certain degree of anonymity and Pseudonymity is also necessary.
- **Proximity and Locality:** This principle tells us that information should be locally preserved. In terms of LTS we can clarify the principle in a way that if information is not disseminated out of a certain locality or area, the likability of tracking will be harder and thus it will be more privacy compliant.

[3] European Commission. 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, November 1995.

3 Identity Metasystem

Identity Metasystem is a notion of abstracting the identity layer from the rest of the system with the intention of gaining interoperability, privacy and usability [7–10]. Identity Metasystem can well be explained using the following analogies: before 1950 programmers had to write assembly codes to fulfill their programming needs. Code pieces written in assembly language required extreme intellectual efforts and more problematically it lacked totally the feature of being portable from machine to machine. Invention of compilers in 1950s and early 1960s brought along an abstraction layer where a generic form of code piece could reside and could be translated back to machine specific assembly code according to need; thus bringing portability to code pieces breaking the machine specific silos. Similarly in the 1970s and early 1980s, before the general adoption of IP, distributed applications were bound to possess network specific knowledge e.g. Ethernet, Token Ring or Frame Relay. But IP was again being an abstraction layer acted as encapsulation of such network specific technologies. Following these two analogies we can think of an Identity Metasystem that acts as such mediator of existing Identity Management systems allowing interoperability and portability of identity [3].

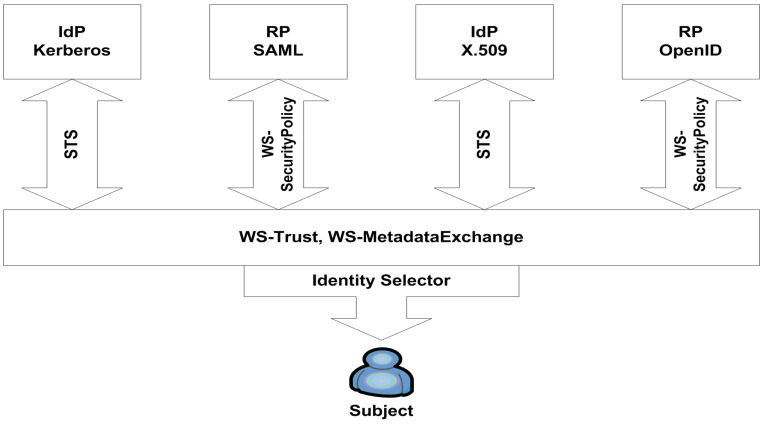


Fig. 2. Identity Metasystem Architecture.

Identity Metasystem is a complex system that integrates the existing IdM systems to provide seamless interoperability and portability. Figure 2 illustrates how IdPs with specific Security Token Services (e.g. Kerberos, X.509 etc.) is federating identity to RPs who accept other types of Security Tokens (e.g. SAML, OpenID). WS-SecurityPolicy is shown to be the policy negotiation mechanism, and WS-trust [11] and WS-MetadataExchange [12] as the abstraction layer.

Having introduced the basic concept of Identity Metasystem in this section, we now address an obvious question that would strike the reader's mind: what is the implication of Identity Metasystem in LTS? In Section 2.3 we examined the principles of privacy in LTS in smart environments. Our objective in this

section is to elaborate on how close to those principles we can bring PAISE by hybridization with Identity Metasystem. First of all, an Identity Metasystem helps users to stay in control of their identity interactions by allowing them to select context relevant digital persona they would like to reveal in a particular instance. These digital personas contain minimum personal information required for a transaction. In case of PAISE the system will only know whether the user is authorized to access a certain resources or not. Secondly, it empowers users to make informed and reasonable decision about disclosing their identity in such environments [13, 14]. Thus, Identity Metasystem can potentially improve privacy in LTS. However, as we have mentioned in Section 2.3, absolute privacy protection is not achievable by the privacy principles from Langheinrich, in a similar way Identity Metasystem will also not provide total privacy protection, rather more privacy awareness.

4 Identity Management Technologies

We have illustrated a conceptual view on Identity Metasystem and its implication in LTS. Before we move into our architectural description of Identity Metasystem in PAISE, in this section we provide an overview of the predominant IdM technologies and compare their compliance with the privacy principles (cf. Section 2.3).

4.1 OpenID

OpenID 1.0 was originally developed in 2005 by Brad Fitzpatrick, Chief Architect of Six Apart, Ltd. OpenID, a protocol for light weighted identity, is adopted by a wide range of websites, especially which have heavy user-generated contents. OpenID Authentication 2.0 [15–17] is a fully backward-compatible open community-driven platform that permits and motivates federated identity. OpenID Authentication 2.0 specification which is a data transfer protocol to support both push and pull use cases. Besides, the community is coming up with extensions to support the exchange of rich profile data and user-to-user messaging. There are three key features of OpenID: Single Sing On, decentralized and light weight identity. According to an article published in German online computer magazine "Heise Online [4]" on 18th January 2008 there were already 370 million OpenIDs existing globally. However, the real number of active OpenID users is still unknown, because big companies like Yahoo and Aol offered an OpenID to all their users, which explains the high number of existing OpenIDs. By Design, the OpenID protocol suffers from serious privacy issues. OpenID allows URL to Identify a subject or a user and the URL that is used to identify the Subject is recyclable. Since OpenID permits URL based identification, it brings the issue of privacy. The privacy of the user using an URL as his

[4] http://www.heise.de/security/Yahoo-will-das-Passwort-Chaos-beenden–/news/meldung/102001, last viewed on Monday, December 01, 2008

OpenID will be compromised somehow the possession of that URL lost. This is how the principles of users' choice, consent, proximity and locality is violated in case of URL recycling.

4.2 Windows CardSpace

Windows CardSpace is a visual metaphor for identity selector for the end-user. Windows CardSpace provides controlling power to the end-users on the fact that which information (about the end-users) should reach to the Relying Party (RP) and which should not. Windows CardSpace is a production of Microsoft shipped with Windows Vista (or as an add-on in Windows XP); it is not meant to replace the other standards handling digital identity rather to utilize and extend them [18]. Windows CardSpace is token agnostic. Microsoft codename "Geneva" is coming up with the next version of Windows CardSpace. "Geneva" is a claim based access platform, which includes three components: "Geneva" Server, Windows CardSpace "Geneva" and "Geneva" Framework [19]. Windows CardSpace has major privacy flaws: firstly it relies on the users' judgments on the trustworthiness of RPs. A CardSpace user is given the freedom to choose one of the options of high-assurance certificate belonging to the RP, ordinary certificate belonging to the RP or RP with no certificates [20]. In terms of the first privacy principle (choice and consent) this certainly gives a lot of power to the user. At the same time the option of allowing RP with no certificates weakens the compliance with the fourth principle (proximity and locality) as information may leak out to an unwanted domain. The second vulnerability is that Windows CardSpace rely on a single layer of authentication. The user has to be authenticated to the IdP using traditional authentication mechanism. If a working session is somehow hijacked or password is cracked, the security of the whole system is compromised. This has been practically showed by two IT-Security students at Horst Grtz Institute for IT Security (HGI), Bochum, Germany where they manipulated the DNS server to implement dynamic pharming attack [5].

4.3 Higgins Trust Framework

Higgins is a software infrastructure that provides a consistent user experience across multiple digital identity protocols, e.g. WS-Trust, OpenID, SAML, XDI, LDAP etc. The main objectives of the Higgins project are the management of multiple contexts, interoperability, and the definition of common interfaces for an identity system. Various technologies including LDAP, SAML, WS-*, OpenID etc. can be plugged into the Higgins framework. The first version, Higgins 1.0 was released in February 2008. The next version, Higgins 1.1 was supposed to be released by June 2009. There are also ideas and concepts in discussion beyond Higgins 1.1 [21].

[5] On the Insecurity of Microsoft's Identity Metasystem CardSpace, Press release, Bochum, Germany, May 27, 2008, http://demo.nds.rub.de/cardspace/PR-HGI-TR-2008-003-EN.pdf

Since Higgins supports various IdM protocols and technologies it inherently takes over the flaws and vulnerabilities of those technologies and protocols. It also does not fourth principle (proximity and locality). However, the combined approach to provide an umbrella framework for IdM allows Higgins users to choose the best combination of technologies suited to their requirements. Moreover, Higgins architecture is most compliant to other three privacy principles (Section 2.3) among the state of the art technologies that have been considered in this evaluation. Therefore, in our architecture we have taken some of the Higgins architectural approach and integrated to our need. In the next section the evaluation result is summarized.

4.4 Evaluation of the State of the Art

Figure 3 shows the summary of our evaluation result. We came up with a scale of poor (-) to very good (++) to show the suitability of the three IdM technologies for getting close to the privacy principles stated in Section 2.3. As we can see, Higgins provides more supports to implement the privacy principles than OpenID and Windows CardSpace. In the following section we will present a generic architecture for Identity Metasystem in PAISE that will be implemented using the Higgins framework in future.

Privacy Principles	OpenID	CardSpace	Higgins
1. Notice	-	+	++
2. Choice and Consent	+	++	++
3. Anonymity and Pseudonimity	+	+	+
4. Proximity and Locality	--	--	--

Fig. 3. A ranking based on our evaluation.

5 Proposed Architecture

Our proposed architecture is based on Identity Federation between three entities: the user, the IdP and the RP. The user requests for access to a certain resource from a RP in the smart environment and gains the access upon required credential is federated by an IdP. Note that same entities can play the role of IdP or RP depending on the context. By means of Identity Federation the proposed architecture will bring more privacy features on top of the PAISE authentication mechanism.

The communication protocol shown in Figure 4 works the following way:

1. The user requests for accessing certain resources.
2. The user is redirected to an IdP by the RP.

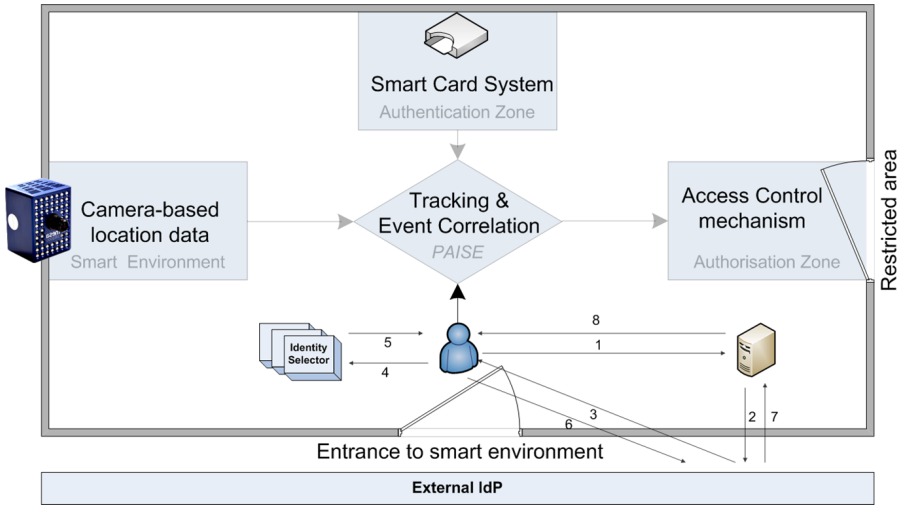


Fig. 4. Communication view point on the PAISE architecture.

3. The IdP requests the user for his credential.
4. The user looks at his InformationCard repository.
5. User selects an InformationCard.
6. The user reveals his credential to the IdP.
7. The IdP releases a token to the RP.
8. The RP grants access to the requested resources. The RP can then use this token to authenticate the user or for some other purposes.

This architecture allows decoupling of user identity from IdP and RP, i.e., users can use different identities and possibly different IdPs towards different RPs in the environment. This is how it captures the essence of being ambient in smart environment and switch identity to plug into different contexts. It leverages location tracking feature of PASIE and identity federation features of Identity Metasystem to enable multi-party federation in an ambient manner.

6 Discussion

In this section we briefly discuss how the proposed architecture accomplishes the privacy principles described in Section 2.3.

– **Notice:** First of all, the user is able to choose an InformationCard as a visual metaphor of his digital identity, which allows him a human readable mechanism to understand exactly what information about him is fed to the system, i.e., the data collection system is more "open" and "transparent" to the user. This exactly reflects the first privacy principle "notice" mentioned in Section 2.3.

- **Choice and Consent:** Secondly, when the user is sending a request for a resource, he is redirected to his IdP and the IdP requests for his credential. In this case the user can choose a suitable digital persona for the given context. This brings the privacy principle "choice and consent" to be applicable in PAISE.
- **Anonymity and Pseudonymity:** Moreover, when the user is federated by an external identity provider to access resources to a smart environment, the system only knows if the user has the right credential to access a certain resources. Obviously, there has to be a predefined trust mechanism between the external IdP and domain of the smart environment the user is accessing. This allows the user a certain degree of anonymity within the area where he is tracked by the PAISE system, i.e., he is not identified as an individual entity rather as an entity belonging to a group.
- **Proximity and Locality:** Finally, his tracking data does not go beyond the locality of his foreign domain, i.e., his IdP who has his actual identification does not gain access to his movement data. Thus it allows the fourth privacy principle "'proximity and locality" to be applicable up to a certain degree.

7 Conclusion & Outlook

To summarize the implication of the "Identity Metasystem" as an add-on to the PAISE model it is important to highlight some benefits of Federated Identity and InformationCard. First of all federated identity makes data collection fragmented, thus leads to less unlinkability without losing the possibility of traceability. Accountability vs. anonymity has been a classic debate between proponents of privacy and security. Federated Identity somehow meets the balance between these two. Another significant addition is an InformationCard. InformationCard provides a clear and usable representation of digital identity. It allows context aware identity selection, empowerment of the user and usability. Although the user is not able to switch off his tracking, by means of identity selector he can choose InformationCard selecting different identities and contexts. This is how PAISE is one step closer to the old definition of privacy given by Brandeis, i.e., the right to be leave one identity mask and select a new one has been accomplished.

References

1. Hansen, M., Kirshmeyer, M., Jensen, C.: Persistent authentication in smart environments. In Proceedings of the 2nd International Workshop on Combining Context with Trust, Security, and Privacy (CAT08) (June, 2008) 31–44
2. Weiser, M.: The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev. **3**(3) (1999) 3–11
3. Langheinrich, M.: Privacy by design - principles of privacy-aware ubiquitous systems. In: UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing, London, UK, Springer-Verlag (2001) 273–291

4. Lockton, V., Rosenberg, R.S.: Rfid: The next serious threat to privacy. Ethics and Inf. Technol. **7**(4) (2005) 221–231
5. Michael, M.G., Fusco, S.J., Michael, K.: A research note on ethics in the emerging age of überveillance. Comput. Commun. **31**(6) (2008) 1192–1199
6. Perusco, L., Michael, K.: Control, trust, privacy, and security: evaluating location-based services. Technology and Society Magazine, IEEE **26**(1) (Spring 2007) 4–16
7. Cameron, K.: The laws of identity. Microsoft Corporation, http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf (2005)
8. Akram, H., Hoffmann, M.: Supports for identity management in ambient environments - the hydra approach. Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on (Oct. 2008) 371–377
9. McLaughlin, L.: What microsoft's identity metasystem means to developers. Software, IEEE **23**(1) (Jan.-Feb. 2006) 108–111
10. Cameron, K., Jones, M.B.: Design rationale behind the identity metasystem architecture (2006)
11. Lawrenc, K., et. al.: Ws-trust 1.3. OASIS Standard, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html (19 March 2007)
12. Ballinger, K., et. al: Web services metadata exchange, version 1.1. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-mex/metadataexchange.pdf (August 2006)
13. Akram, H.I., Hoffmann, M.: User-centric identity management in ambient environments. International Journal on Advances in Intelligent Systems **vol 2, no 1** (year 2009) 254 – 267
14. Akram, H., Hoffmann, M.: Laws of identity in ambient environments: The hydra approach. In: UBICOMM '08: Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Washington, DC, USA, IEEE Computer Society (2008) 367–373
15. Miller, J.: Yadis 1.0. http://yadis.org/papers/yadisv1.0.pdf (March 2006)
16. Recordon, D., Reed, D.: Openid 2.0: a platform for user-centric identity management. In: DIM '06: Proceedings of the second ACM workshop on Digital identity management, New York, NY, USA, ACM (2006) 11–16
17. Recordon, D., Fitzpatrick, B.: Openid authentication 1.1. http://openid.net/specs/ (2006)
18. Mercuri, M.: Beginning Windows CardSpace: From Novice to Professional. Apress, Berkely, CA, USA (2007)
19. Brown K, Mani, S.: Microsoft code name "geneva" framework whitepaper for developers. Microsoft Corporation, http://download.microsoft.com/download/7/d/0/7d0b5166-6a8a-418a-addd-95ee9b046994/GenevaFrameworkWhitepaperForDevelopers.pdf (2008)
20. Alrodhan, W.A., Mitchell, C.J.: Addressing privacy issues in cardspace. In: IAS '07: Proceedings of the Third International Symposium on Information Assurance and Security, Washington, DC, USA, IEEE Computer Society (2007) 285–291
21. Ruddy, M., Trevithick, P., Nadalin, T., Olds, D.: Higgins trust framework. Digital ID World (2006)