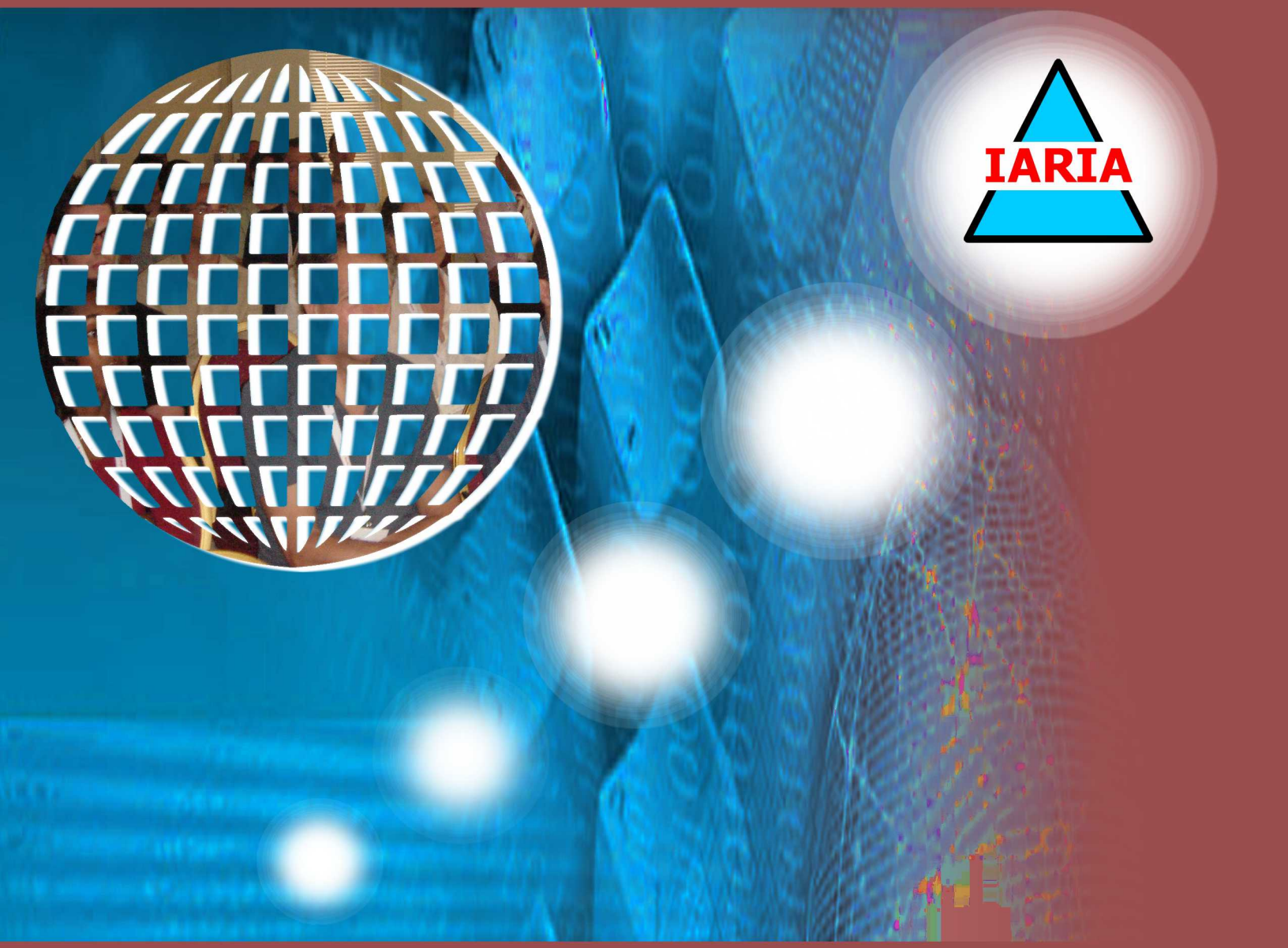
International Journal on Advances in Security



2008 vol. 1 nr. 1

The *International Journal On Advances in Security* is Published by IARIA.
ISSN: 1942-2636
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal On Advances in Security
Volume 1, Number 1, 2008

Editorial Board

First Issue Coordinators

Jaime Lloret, Universidad Politécnica de Valencia, Spain
Pascal Lorenz, Université de Haute Alsace, France
Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada

Quantum Security

- Marco Genovese, Italian Metrological Institute (INRIM), Italy
- Masahito Hayashi, Tohoku University, Japan
- Vladimir Privman, Clarkson University - Potsdam, USA
- Don Sofge, Naval Research Laboratory, USA

Emerging Security

- Nikolaos Chatzis, Fraunhofer Gesellschaft e.V. - Institute FOKUS, Germany
- Rainer Falk, Siemens AG / Corporate Technology Security - Munich, Germany
- Ulrich Flegel, SAP Research Center - Karlsruhe, Germany
- Matthias Gerlach, Fraunhofer FOKUS , Germany
- Stefanos Gritzalis, University of the Aegean, Greece
- Petr Hanacek, Brno University of Technology, Czech Republic
- Dan Harkins, Aruba Networks, USA
- Dan Jiang, Philips Research Asia – Shanghai, P.R.C.
- Reijo Savola, VTT Technical Research Centre of Finland, Finland
- Frederic Stumpf, Technische Universitat Darmstadt, Germany
- Masaru Takesue, Hosei University, Japan

Security for Access

- Dan Harkins, Aruba Networks, USA

Dependability

- Antonio F. Gomez Skarmeta, University of Murcia, Spain
- Bjarne E. Helvik, The Norwegian University of Science and Technology (NTNU) – Trondheim, Norway
- Aljosa Pasic, ATOS Origin, Spain
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Michiaki Tatsubori, IBM Research - Tokyo Research Laboratory, Japan
- Ian Troxel, SEAKR Engineering, Inc., USA

- Marco Vieira, University of Coimbra, Portugal
- Hans P. Zima, Jet Propulsion Laboratory/California Institute of Technology - Pasadena, USA // University of Vienna, Austria

Security in Internet

- Evangelos Kranakis, Carleton University, Canada
- Clement Leung, Victoria University - Melbourne, Australia
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Yong Man Ro, Information and Communication University - Daejon, South Korea

International Journal On Advances in Security
Volume 1, Number 1, 2008

Foreword

Finally, we did it! It was a long exercise to have this inaugural number of the journal featuring extended versions of selected papers from the IARIA conferences.

With this 2008, Vol. 1 No.1, we open a long series of hopefully interesting and useful articles on advanced topics covering both industrial tendencies and academic trends. The publication is by-invitation-only and implies a second round of reviews, following the first round of reviews during the paper selection for the conferences.

Starting with 2009, quarterly issues are scheduled, so the outstanding papers presented in IARIA conferences can be enhanced and presented to a large scientific community. Their content is freely distributed from the www.iariajournals.org and will be indefinitely hosted and accessible to everybody from anywhere, with no password, membership, or other restrictive access.

We are grateful to the members of the Editorial Board that will take full responsibility starting with the 2009, Vol 2, No1. We thank all volunteers that contributed to review and validate the contributions for the very first issue, while the Board was getting born. Starting with 2009 issues, the Editor-in Chief will take this editorial role and handle through the Editorial Board the process of publishing the best selected papers.

Some issues may cover specific areas across many IARIA conferences or dedicated to a particular conference. The target is to offer a chance that an extended version of outstanding papers to be published in the journal. Additional efforts are assumed from the authors, as invitation doesn't necessarily imply immediate acceptance.

This particular issue covers papers invited from those presented in 2007 and early 2008 conferences. The papers cover mostly issues pertaining to user-centric aspects on identity in digital eco-systems and mechanism to handle identity semantics; complementarily, enhanced security is presented via redundancy and adaptability.

We hope in a successful launching and expect your contributions via our events.

First Issue Coordinators,
Jaime Lloret, Universidad Politécnica de Valencia, Spain
Pascal Lorenz, Université de Haute Alsace, France
Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada

International Journal On Advances in Security
Volume 1, Number 1, 2008

CONTENTS

A Distributed Identity Handling Approach Enriched with Identity Semantics

Mohammad M. R. Chowdhury
UniK-University Graduate Center
Post Box. 70, 2007 Kjeller, Norway
mohammad@unik.no

Josef Noll
UniK-University Graduate Center
Post Box. 70, 2007 Kjeller, Norway
josef@unik.no

Abstract

People rely on many forms of identities to access offline and online services. The inconvenience of possessing and using identities creates significant security vulnerability. This paper proposes an identity handling mechanism enriched with identity semantics which is believed to ease and secure identity usage. In this regard, user's identities are classified into personal, corporate and social identities, and they are going to be distributed over user's personal device and a secure network place. Corporate and social identities are represented through user's roles and relationships exploiting Web Ontology Language. Secure service access is ensured through multi-factor authentication method. Access is further restricted through authorization making use of user's defined roles and relationships. This paper demonstrates an implementation of service access by means of authentication and authorization through one's personal and corporate or social identities. The proposed solution is analyzed and compared with other relevant concepts, methods and solutions.

Keywords: authentication, authorization, identity, ontology, role, relationship

1. Introduction

Identification is a process through which a system ascertains the identity of a person who is trying to gain access to the system. It is essential to provide access to various value added services. Human beings play different roles while interacting with these services. Paper-based identities cannot be used while accessing services in the digital world. Moreover, different types of services require different types and forms of identities. People increasingly use computers to do business over the Internet. But accessing online value added services invariably requires typing various usernames and passwords for identification. These passwords can be captured and reused by hostile parties. To make the service access simple, hassle-free and above all secure, a manageable but usable identity mechanism is expected.

Mobile phone penetration is expected to reach 100% in most of the European countries[1]. It has become a foremost electronic device for worldwide communication because of its mobility, seamless and secure access provision to networks. In addition to this, mobile phone has always online functionality. Lately, computing capabilities of the mobile devices are enhanced manifold. Nowadays, there are provisions for being connected with the Internet using SIM from the laptop computers. It is evident that ubiquitous access and pervasive computing facilitate service access anywhere, anytime. In this paper, we focus on accessing the Web services through Mobile Phone/SIM card authentication.

User's identity data is not merely 'Information'. The semantics of identity information and the data itself are crucial for decision making, such as deriving access authorization decisions. This paper extends such interpretation towards an identity handling mechanism. In this regard, user's identities are classified into personal, corporate and social identities, and they are going to be distributed over user's personal device and a secure network place. User's roles and relationships represent a part of the corporate and social identities exploiting Web Ontology Language (OWL), a semantic technology standard for knowledge representation. Security in service access is ensured through multi-factor authentication method. Authenticated access is further restricted through authorization making use of user's defined roles and relationships. This paper demonstrates an implementation of secure service access by means of authentication and authorization through personal and corporate or social identities.

The paper starts with the definitions of identity and its management emphasizing the motivation of terming roles and relations as identity of users (section 2). Section 3 introduces semantic technologies and discusses the motivation behind the use of them. The paper then illustrates

[1] Telecom & IT research reports by RNCOS, *European Mobile Market Scenario to 2012*, `http://www.rncos.com/Report/IM101.htm` [retrieved on Jan. 17, 2009]

the generic architecture of the proposed identity handling mechanism in section 4. Section 5 addresses the security requirements and methods of the proposed identity mechanism bringing the detail authentication and authorization aspects. Section 6 presents the concept of service interaction using the distributed identity mechanism and shows the prototypical implementation of the presented concept. The paper will review some of the related works and then provide critical analysis on different aspects of the proposed distributed identity mechanism in section 7 and 8. The paper concludes with a summary of the paper and comments on future research.

2. Identity

In this section, we look for the definitions of identity from different angles and explain how these definitions motivate us to extend the social aspects of identity with a goal to interact services securely.

2.1 Definition of Identity

In social sciences, Identity is broadly used to describe an individual's comprehension of him or herself as a discrete, separate entity[2]. Analyzing the current usage of identity in ordinary language and social science discourse, it can be summarized that identity is currently used mainly in two linked senses, 'social' and 'personal' [10]. In the former, a person is distinguished by rules deciding membership and characteristic features or attributes. In the second sense, identity is distinguishing characteristics that a person takes a special pride in. 'How people relate to others' is termed as identity by [15]. It 'refers to the ways in which individuals and collectivities are distinguished in their social relations with other individuals and collectivities' [19]. According to Dick Hardt, CEO of Sxip Identity, identity is also what i prefer, what my interests are, what my roles are in real life [14].

In this paper, we take into account both the 'personal' and 'social' sense of identity. 'Personal' sense is realized through possessing or knowing some identifying characteristics and the later is established through roles and relationships of an individual.

2.2 Role as Identity

Central to the identity theory developed by Stryker [32], McCall and Simmons [24], and Turner [34] is the concept of role identities. The Theory links self attitudes, or identities, to the role relationships and role-related behavior of individuals. The theorists argue that the self consists of a collection of identities, each of which is based on occupying a particular role. Stryker said 'the number of identities is limited only by the number of structured role relation ships one is involved in' [32].

2.3 Relationship as Identity

In the Social Identity Theory [33], beyond the 'personal self' a person has several selves that correspond to widening circles of group membership. Apart from 'personal self', an individual has multiple social identities. Social identity is the individual's self-concept derived from perceived membership of social groups [16]. The group membership creates positive self-esteem by positively differentiating their ingroup from a comparison outgroup on some valued dimension. Because of these, people's sense of who they are is defined in terms of 'we' rather than 'I'. Thus the social identity differs from the notion of personal identity which refers to self-knowledge that derives from the individual's unique attributes.

2.4 Paper-based Identity

It is the traditional form of identity. People are carrying a good number of paper-based identities, for example, passport/personal ID, credit cards, bank cards, student card, office ID, driving license etc. with them. Nowadays, people increasingly use smart cards with electronic chip for service access and payment. It enhances the security and allows storage of user details on the card. These are normally used at designated service points that can recognize specific smart cards. The possession factor of such identities are coupled with a knowledge factor like PIN codes, which authenticate the true owner and serve as additional security requirements (this phenomena will be discussed more detailed in section 5). However, with the increasing digitization of identity transaction, many of the paper-based identities are gradually replaced or supplemented by digital identities.

2.5 Digital Identity

Digital identity is the digital representation of a set of claims made by one digital subject about itself or another digital subject. A digital subject can be human or non-human. Instead of set of claims made by parties, digital identity can also be defined as a collection of information that relates to an individual, that is created and managed as a single unit in a network, and that is stored in electronic form[3].

[2]Identity(social science) http://www.answers.com/topic/identity-social-science [retrieved on Jan. 18, 2009]

[3]Definition of digital identity, http://idcorner.org/2005/03/07/on-the-definition-of-digital-identity/ [retrieved on Jan. 18, 2009]

2.6 Identity Management

In information systems, identity management is a broad administrative area that deals with the management of identity life cycle of entities starting from establishing identities and ending with repealing those identities, when required. This is the pure identity paradigm that does not consider access or entitlements. Within the life cycle, identities are described through various attributes. Identity management involves user access paradigm which considers the management of identity associated data required to access a system. User may know these data beforehand or a secure physical device may contain these. Here, entities are identified by presenting these identity associated data. The access to resources within the system can be controlled by associating user roles, rights and restrictions, presence with the established identity. Thus system can deliver personalized services. This is service paradigm perspective of identity management. Therefore, in addition to identity and access manager, identity management solutions[4] [5] also encompass role manager.

In this paper, we are concerned with user access and service paradigm identity management. That is why, we also demonstrated how the proposed distributed identities can facilitate service interaction.

3 Introduction to Semantic Technology

We used semantic technologies to represent part of user's corporate and social identities, and to realize access authorization decisions (detail in section 5). This section introduces the technologies and the motivations of using them in this work.

Semantic Web [4] provides various technologies to capture the knowledge about a domain of interest in the form of concepts and their relationships at different levels of abstraction. It supports the reasoning about both the structures and the properties of the elements that constitute the system. Ontologies [11] are the cornerstone technology of Semantic Web. Among the different ontology languages, the OWL [31] is chosen because it facilitates greater machine interpretability of the Web content than that is supported by XML, RDF, and RDFS by providing additional vocabularies along with formal semantics. RDF/XML syntax is the basis of serialization in OWL ontology. There are three species of OWL: OWL Lite, OWL DL and OWL Full and these are designed to be layered according to their increasing expressiveness.

[4]Oracle Identity Management, `http://www.oracle.com/products/middleware/identity-management/identity-management.html` [retrieved on Jan. 18, 2009].

[5]Sun Identity Management, `http://www.sun.com/software/products/identity/offerings.jsp` [retrieved on Jan. 18, 2009].

Apart from the representation of domain knowledge, the architecture requires more expressivity to deduce decidable conclusions which in fact provide the authorization decisions during controlling access to a system. To enhance the expressivity of the ontology, we decided to use OWL DL which is based on the Description Logics (DL) and amenable to automated reasoning[6]. Though OWL DL lacks in expressivity power compared with OWL Full, it maintains decidability[7] and computational efficiency. The computational efficiency is important since the scheme has to handle many relations.

As the expressivity provided by the OWL is limited by tree like structures [25], the implicit relations representing the restricted access scenarios cannot be inferred from the indirect relations between the entities. These require rule support and interworking with ontologies. One suitable rule language is the Semantic Web Rule Language (SWRL) [17]. Along with SWRL, we also use Semantic Query-Enhanced Web Rule Language (SQWRL[8]) to further enhance the expressivity of OWL, specifically to derive the access authorization decisions based on defined knowledge (ontologies).

4 Generic Architecture of the Identity Handling Mechanism

This chapter introduces the generic architecture of the identity handling mechanism and illustrates each of its components.

4.1 Roles in Life

Every human being plays numerous roles in life to live. As a student, we are attending an educational institute; as a researcher or engineer, we are working in a company; as a consumer, we are buying things with cash or credits; we are maintaining social relationships with family, relatives, neighbors and colleagues. While exercising these roles in life, we are interacting with many service providers to receive different types of services. For example, as a student one has access to various services of the institute. Analyzing these scenarios, it can be said that every human being plays roles basically in three different areas, personal, professional and social areas. Therefore in reality, leading everyday life is nothing but playing some personal roles, professional roles and social roles in general.

[6]Reasoning is the process of deducing implicit or indirect relations from the explicit knowledge.

[7]Logics are decidable if computations/algorithms based on the logic will terminate in a finite time.

[8]Semantic Query-Enhanced Web Rule Language (SQWRL), `http://protege.cim3.net/cgi-bin/wiki.pl?SQWRL` [retrieved on Jan. 18, 2009]

In this article, we are proposing a concept of 'My digital identity' which contains 'My personal identity (PID)', 'My corporate identity (CID)' and 'My social identity (SID)' that would represent ourselves and our relevant real life roles to the digital world. 'My personal identity' can be used to identify ourselves in our personal and commercial interactions. Similarly, 'My corporate identity' and 'My social identity' can be used in our professional and interpersonal interactions respectively.

4.2 Personal, Corporate and Social Identities

Each of these three identities will have several identifiers. Each identifier will be used to access several relevant services and a number of attributes will characterize an identifier (see figure 1). Attributes are those set of characteristics of an identifier that are required by the service providers during service interactions. For example, passport can be one of the identifiers and name, date of birth, date of issue, date of expiry, the country that issued the passport, passport number etc. can be its attributes. The passport

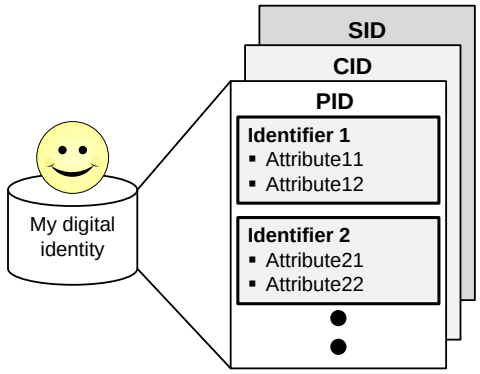


Figure 1. Generic architecture of 'My digital identity'.

that is in fact the most important personal identity issued by the government and used to deal with many government or non-government services. Similarly, another identifier will be used to get access to financial services, like, buying something through credit cards. Attributes of such identifiers are name of the person who holds the credit card (may be optional), number of the card, pin code, date of expiry etc. My PID might have some more identifiers to access our home premises, home network or VPN etc. In the same way, My CID and My SID will have several such identifiers and attributes. My CID might hold the identifiers to access our office premises, office LAN/VPN etc. According to Dick Hardt, individual's interests, fondness, preferences or tastes are also part of his/her identity [14]. In the proposed identity model, these features will also be dealt with by My CID and SID through roles and relations. It may also include identifiers for accessing my email, messenger, IP telephony etc. Each identifier will contain only the required identifying information that a service provider needs to know. 'My digital identity' thus, ensures the minimum disclosure of identifying information.

4.3 Representation of Relationship

Following the research in social science, we proposed the notion of social identity, which in this paper means individual's relationship to a group or with other individuals. Social relation can refer to a multitude of social interactions, regulated by social norms, between two or more people, with each having a social position and performing a social role. We consider the fact that, in the social context we sometime like to be identified as 'member of Cycling Group' or 'Friend of X' etc.

In our research, we represented these social identities using ontologies and we used OWL to design these ontologies. Later (in section 5.3), it is explained how these identities are exploited to authorize a person to see and access group's resources.

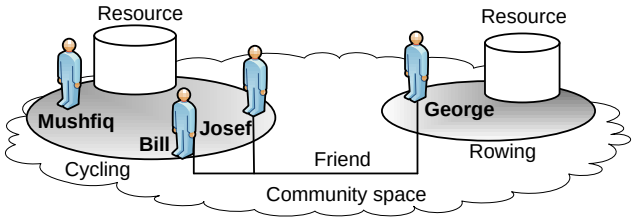


Figure 2. A sample community structure.

Suppose, there exists a community space in the network where there are two communities: Cycling and Rowing, each containing community and public resources. Bill, Josef and Mushfiq are members of Cycling community, while George is a member of Rowing community. Bill, George and Josef are friends to each other. Figure 2 illustrates this sample community structure. The relationship to the community (membership) and the relationship among individuals are exploited to provide access authorization to the right resources. A virtual social network may contain such architecture to ensure security and privacy of community itself and its members. Figure 3 shows the sample codes in RDF/XML (OWL syntax) representing a community environment containing members, some of whom are friends of each other, while some are not.

```
-----
-----
<Community rdf:ID="Cycling">
<hasMember rdf:resource="http://www.owl-ontologies.com/OntologyBill4.owl#Bill"/>
<hasMember rdf:resource="#Mushfiq"/>
<hasMember rdf:resource="#Josef"/>
<hasCommunityResource rdf:resource="http://www.owl-ontologies.com/
OntologyBill4.owl#CyclingPartyVideo"/>
<hasPublicResource rdf:resource="http://www.owl-ontologies.com/
OntologyBill4.owl#HowToCycleVideo"/>
</Community>
<Community rdf:ID="Rowing">
<hasMember rdf:resource="#George"/> </Community>
-----
-----
<owl:Class rdf:ID="Member"/>
<Member rdf:ID="Bill">
<belongTo rdf:resource="#Cycling"/>
<hasFriend rdf:resource="#George"/>
<hasFriend rdf:resource="#Josef"/>
<hasPrivateResource rdf:resource="http://www.owl-ontologies.com/
OntologyBill4.owl#PrivatePartyVideo"/>
</Member>
<Member rdf:ID="Josef"> <belongTo rdf:resource="#Cycling"/>
</Member>
<Member rdf:ID="Mushfiq">
<belongTo rdf:resource="#Cycling"/> </Member>
<Member rdf:ID="George"> <belongTo rdf:resource="#Rowing"/>
</Member>
-----
-----
```

Figure 3. RDF/XML code sample representing a community environment.

4.4 Representation of Role

Individuals gain a social identity and group identity by their affiliation. Individuals play numerous roles in the institutions they are associated with. We distinguish professional institutions from the other social institutions, because the former demands the higher security requirements in participation than the later. Therefore, in this paper we mostly focused on roles an individual plays in professional organization. In this regard we are going to illustrate a specific scenario. Nowadays people in business organizations increasingly work in project oriented environments. The project members come from different departments. The role of each department constitutes at least a supervisor and subordinate employees. Similarly, each project contains a project leader and members. From the sense of identity, we are accustomed to identity ourselves as 'supervisor of department X' or 'project member of project Y', especially in social context. These distinctions of roles play a crucial role in controlling the access to work unit's resources. For this purpose, we also represented this specific scenario of roles using OWL.

Suppose, an organization (a company) consists of two departments: Dept. A and B, each containing some resources. Hans Christian is the supervisor of Dept. B and Josef Noll plays the similar role in Dept. A. Erik Swansson and George Kalman are the employees of Dept. B and A respectively. Figure 4 shows such a sample organizational

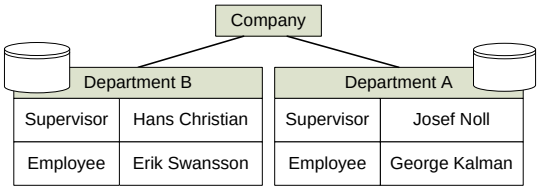


Figure 4. A sample organizational structure.

```
-----
-----
<Department rdf:ID="DepartmentA">
<hasResource rdf:resource="#DeliverableDeptA"/>
<hasResource rdf:resource="#DocDeptA"/>
<hasResource rdf:resource="#AdminResDeptA"/> </Department>
<Department rdf:ID="DepartmentB">
<hasResource rdf:resource="#DeliverableDeptB"/>
<hasResource rdf:resource="#DocDeptB"/>
<hasResource rdf:resource="#AdminResDeptB"/> </Department>
-----
<EmployeeID rdf:ID="Erik_Swansson">
<hasRole rdf:resource="#DeptB_Employee"/> </EmployeeID>
<EmployeeID rdf:ID="George_Kalman">
<hasRole rdf:resource="#DeptA_Employee"/> </EmployeeID>
<EmployeeID rdf:ID="Hans_Christian">
<hasRole rdf:resource="#Supervisor_Hans"/> </EmployeeID>
<EmployeeID rdf:ID="Josef_Noll">
<hasRole rdf:resource="#Supervisor_Josef"/> </EmployeeID>
<Supervisor rdf:ID="Supervisor_Hans">
<rolePlaysIn rdf:resource="#DepartmentB"/> </Supervisor>
<Supervisor rdf:ID="Supervisor_Josef">
<rolePlaysIn rdf:resource="#DepartmentA"/> </Supervisor>
<Dept_Employee rdf:ID="DeptA_Employee">
<rolePlaysIn rdf:resource="#DepartmentA"/> </Supervisor>
<Dept_Employee rdf:ID="DeptB_Employee">
<rolePlaysIn rdf:resource="#DepartmentB"/> </Supervisor>
-----
-----
```

Figure 5. RDF/XML code sample representing an organization.

structure. We will see later (in section 5.3) that depending on roles and relationships to the work units, each individual is authorized to access the right resources. The sample codes in RDF/XML are shown in figure 5.

4.5 Distributed Identity Mechanism

The paper introduces a notion of distributed identity and this section explains the mechanism in detail. With the identity services subscription, Identity Provider (IDP) issues a certificate to the user and allocates a secure identity space in the network. User identity data and attributes are distributed and stored into two places. A part of the user identities that contain very sensitive user information like, PID (e.g., creditcard identifier, home admittance key) will be stored (permanently or temporarily) in the SIM card of mobile phone. This is the primary part of the identities which is used to access the remaining of identities. Therefore, access to these requires strict authentication. The other part of user identities which need medium/low authentication requirements,

Table 1. The PID, CID and SID, their realization, storage location and security requirements.

Identity	Examples	Realization	Location	Security requirements
PID	PID (Credit card)	Certificate + key (public + private)	SIM	High
	PID (Home admittance)	Fixed binary key	SIM	High
CID	CID (Office admittance)	Fixed binary key	SIM	High
	CID (Office admittance)	Temp. binary key	Network	Medium
	CID profile (Role)	foaf/OWL (RDF+XML)	Network	Medium
SID	SID profile (Relation)	foaf/OWL (RDF+XML)	Network	Medium/Low

for example social identities and preferences (SID), will be stored into the secure identity space in the network. Table 1 gives several examples of PIDs, CIDs and SIDs, their possible realizations and where these identities will be located or stored. Considering the various levels of security, the corresponding security requirement of each identity is also mentioned.

During the subscription, an operator can load the SIM card with a private key for the user. PID credentials are realized through certificates and keys provided from the Banks. These can also be stored in the SIM card. The trusted third party (whoever it is) can mediate the whole process. There are few possibilities of realizing PID for home admittance. Admittance keys can directly be stored in binary format in the SIM card or a hash can be generated from the stored private key and hash algorithm. The keys or a hash can later be transferred to other devices using NFC technology. CID and SID profiles and preferences are realized using either FOAF (friend of a friend) or Web Ontology Language (OWL) and stored these foaf/owl files in the network. We think Semantic Web Technology (foaf/OWL) can provide solutions to access control and privacy in corporate and social environment by defining users' roles, relations and access and privacy policies. We have already represented such provisions in OWL implementing identity handling, access control and privacy support in corporate and social community areas in a separate work [8]. Detail description of the work is beyond the scope of this paper. However, we introduce the motivation of using Semantic Technologies for the purpose of security and privacy assurance in section 3

4.6 The Role of Identity Providers

The role of an identity provider is very crucial in identity provisioning. Similar to the subscription of voice and data services, access to identity services subjects to explicit agreement between users and identity providers. An IDP is maintaining strong trust relationships among the subscribers, service providers and the other IDPs. Identity providers may come from users social, corporate or personal domain. Security requirements of identity provisioning from these domains depend on the relevant service access security demands. State/government is the traditional and most accepted identity provider in national and international level providing citizen ID. With strong regulations in place (by state), banks and mobile operators can also act as an IDP. Having a state or citizen ID is obligatory to establish access to some of these services. Several service access (e.g., access to Banks or Creditcards) requires high security environments and therefore the roles of these IDPs are strictly regulated.

4.7 SIM as Identity Storage Element

The SIM card is considered as secure identity storage place because it can be revoked, there are possibilities of further security enhancements in it and user now-a-days can rarely be found without a mobile phone. High capacity SIM cards are available in the industry with increased memory size, additional cryptographic and high speed communication (SIM-handset, SIM-network) capabilities [35]. Handling identities from the SIM gives the user control over the usage of his identities. It is expected that IDPs do not own or control SIM card rather act as facilitators to manage identities. To manage multiple credentials, IDPs can load additional IDs confidentially to either a SIM card (with over-the-air provisioning) or at network identity space with user's consent. In case of loosing the SIM card, a new one can be ordered and the identities previously stored in the card can be reloaded.

With the identity subscription certificate user can identify himself to access the network identity repository that contains identities for example SIDs. These identities will be used to access services that need medium or low level of security requirements. The SIM card holds only the most sensitive user identities. As the network is vulnerable to many security threats, only information of less sensitive character are stored in the network.

5 Security Mechanism

This section discusses the security mechanism introducing the levels of security and explaining the authentication and authorization methods in detail.

5.1 Levels of Security

Ensuring secure service access using proposed identity mechanism is a challenging issue, considering the fact that we are going to store part of the identity in the network that is vulnerable to electronic attack. It has been proposed in this paper that the mobile phone will act as the primary device to access 'My digital identity' in the network. In addition to this, a part of the identity that requires the highest security will be stored in mobile phone SIM card. Here, it is assumed that the user has the provision for 'always-on' functionality in his mobile phone.

Access control is a general way to ensure security in accessing services and resources in the Web. It mainly includes Authentication and Authorization. These two meet two different security requirements. The former verifies user's identity and the authorization assures users rights in a system. Moreover, multi-factor authentication enhances the security of authentication mechanism.

Different levels of authentication mechanisms need to be maintained depending on service access security requirements. From user point of view, a securely maintained communication channel is required to exchange very sensitive user information with the service provider. There are services that require only little information about the user. Highly secured infrastructure is not a necessity for them. Besides, building or maintaining very secure channel requires good investment as well. Therefore, different levels of security should be employed for different types of services. Analyzing all these aspects, [27] introduced three levels of security: Nice to know, need to know, have to know according the increasing security requirements (see figure 6).

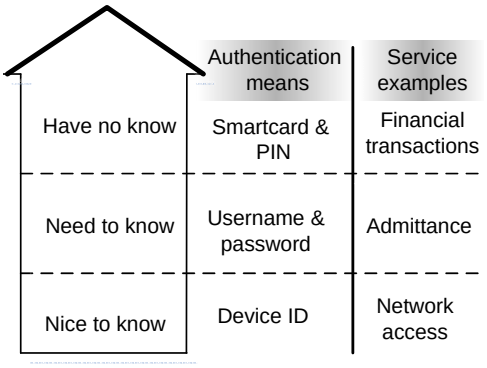


Figure 6. Levels of security based on security requirements of services.

5.2 Authentication

Authentication is the process of identifying an individual. It gives individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about access rights of individuals.

5.2.1 Multi-factor Authentication

An authentication factor is the information or the process used to authenticate or verify the identity of an entity requesting access to a system under security constraints. There are generally three types of authentication factors (listed from weakest to strongest): something a user knows (e.g., password), something a user has (e.g., security token), and something a user is or does (e.g., biometrics). The process of combining multiple authentication factors is called multi-factor authentication. Single factor authentication can be compromised quite easily. Hence, multiple factors can make the authentication stronger.

In the proposed system, we also include multi-factor authentication mechanism. Through a nice to know authentication method, user can access 'My digital identity' and, through a need to know authentication mean, user can access most other services, such as, accessing messenger (msn or yahoo), my address book, voip services, e-mail account; accessing home or office premises etc. Nice to know services are network access, where knowledge about usage is only required. Access to a very personal device like, SIM of a mobile phone (it contains unique identifier), can provide nice to know authentication to the network. Need to know services have higher security requirements. In addition to the device identifier, these require passwords or PIN. If the mobile phone usage is PIN protected, need to know authentication can be avoided. Highest security requirements are needed for have to know services. Users have to be authenticated through a have to know authentication mechanism to use the identifiers that are necessary to access financial services such as, banking, online shopping etc. Here, we are proposing to deploy the have to know authentication mechanism in SIM card through Public Key Infrastructure (PKI). Thus SIM card will be a part of 'My digital identity'. This will significantly minimize the possibility of disclosure of identities for financial services, in case there are electronic attacks on network contents of 'My digital identity'.

5.2.2 Extended SIM Card Authentication

Currently, the SIM card provides the nice to know access to network. We propose that the SIM card authentication will also be enough to enter the part of 'My digital identity' located in the network. The higher security requirements that

need to know services may require might also be satisfied through SIM card authentication [27].

As proposed, the have to know authentication mechanisms will be realized in SIM card. Hence, we are introducing an extended SIM card (ESIM) is a customized variant of USIM (Universal Subscriber Identity Module) and it has the capability to hold multiple credentials. One will be responsible to provide the nice to know and need to know authentications and another one will store the have to know authentication mechanisms. ESIM will also be a part of 'My digital identity' which the users will always carry.

The have to know authentication mechanism in SIM card can be realized by implementing PKI. PKI enables users to secure a public network (e.g., Internet) through using public and private cryptographic key pair that is obtained and shared through trusted authority (Certificate Authority) [13]. In mobile networks, there exists a formal relationship between users/subscribers on one hand and the network operator of the other. Therefore, network operator can naturally play the role of Certification Authority (CA). The users private key as well as the root CA public key can be distributed in a secure way in the form of SIM card. The formal relationship, which the operators already have through roaming agreement, could be extended to cross-certification of each other public keys. Mobile network operators therefore are in a very strong position to establish themselves as CAs, and the mobile device naturally lends itself to become a secure storage medium for these cryptographic keys [20].

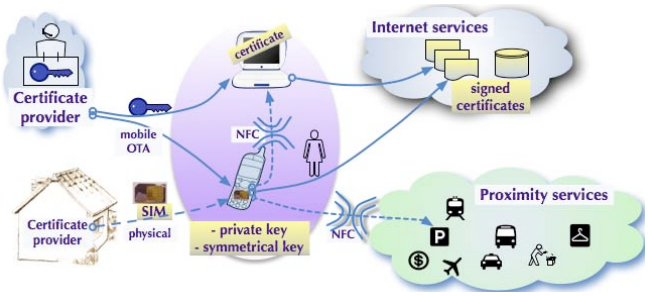


Figure 7. SIM based certificate and key handling.

PKI enables the parties in a dialogue to establish confidentiality, message integrity and user authentications without having to exchange any secret information in advance. Figure 7 illustrates SIM based certificate and key handling provisions. Certificates can be provided to the mobile phone either physically or through mobile over-the-air (OTA)[9] provisioning. It is possible to sign the certificate/transactions using stored private key from the mobile phone or PC. These signed certificate/transactions will provide authentication, integrity and non-repudiation services during service interactions.

[9]Over-the-air (OTA) is a standard for the transmission and reception of application-related information in a wireless communications system.

5.2.3 Acceptability of Mobile Phone as Identifier

Nowadays mobility of people increases due to dynamic life style and working nature. The mobile phone has become a foremost electronic device not only for communication but also for managing different other activities, such as, banking, collecting information from web, checking emails etc. Mobile phone penetration is expected to reach 100% in most of the developed countries. So, the basic infrastructure to use the mobile phone as identifier is already in place. Currently, different types of access systems can be found in wireless networks. Services are expected to be interoperable in different wireless communication systems. A SIM is a temper resistant device in a wireless system holding subscriber identity and authentication information. The SIM card in the mobile phone has the capability to provide all levels of authentication, and support mechanisms for revocation of credentials stored in the SIM card [26]. It is only active if authenticated by the network operator. If it gets stolen, the operator can disable the card. SIM card opens for authentication and encryption in every wireless network (Bluetooth, WLAN, WiMAX) in addition to GSM and UMTS [26]. So, SIM card enables authentication mechanism to interact different services will certainly give a technological edge to the development of future wireless technologies and services. By storing a part of the identity in the network, we are reducing the volume of data transfer from mobile phone to network. In consequence, the additional data transfer due to the use of such system will leave a very little effect on the capacity of air interface. Therefore, the acceptabilirty of mobile phone as identifier is expected to be very high.

5.3 Authorization

When a user is authenticated, information system has to make sure that he accesses only what he is allowed to. Access authorization determines a) what an user can do in a system (access to contents/services) and b) with which privileges? (e.g., read and/or write over the contents).

5.3.1 Authorization Based on Relationships

In this section, we refer to the representation of identity (relationship) to discuss the authorization based on relationships. Access authorization decisions are derived based on the relationship among the individuals, and between the individuals and the community they belong to. Bill, Josef and Mushfiq belong to the Cycling community; George belongs

to the Rowing community; Bill has two friends: George and Josef, and he also possesses a private resource: Private Party Video. Cycling community has community and public resource. From this scenario, we are expecting the following access situations:

- Cycling community members (Bill, Josef and Mushfiq) can access community resource: Cycling Party Video with full access privilege (streaming and download the full length).
- Josef is expected to get full access to Bill's private resource: Private Party Video, as he is not only a friend of Bill, but they also belong to the same community.
- George gets limited access (preview only) to Cycling Party Video as he is not a member of Cycling community.
- George is allowed to see the Private Party Video of Bill with limited access privilege though he is a friend of Bill, as they are not in the same community.
- Mushfiq can see the Private Party Video of Bill with limited access privilege as he is not a friend of Bill, though they belong to the same community.

These access situations are realized through a set of access authorization policies and they are designed through rules exploiting SWRL. Figure 8 illustrates these policy

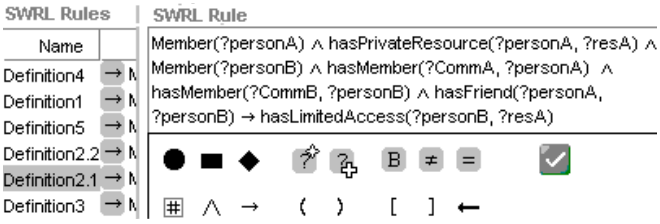


Figure 8. Access policies are designed through rule using SWRL.

sets and a snapshot of one of the rules. The rules executed using Jess rule engine to derive the access authorization decisions. Figure 9 shows the authorization decisions which follow the defined access situations. The access privileges are not shown in the figure but these are working from the back end.

5.3.2 Authorization Based on Roles

In section 4.4, we have represented an organizational environment containing different work units, employees, their roles and relationships with relevant work units. Table 2 shows the detail organization structure containing roles and

Person	Access to Resource
Bill	CyclingPartyVideo1
Josef	CyclingPartyVideo1
Mushfiq	CyclingPartyVideo1
Josef	PrivatePartyVideo1
George	CyclingPartyVideo1
George	PrivatePartyVideo1
Mushfiq	PrivatePartyVideo1

Figure 9. Access authorization decisions derived executing SWRL rule shown in figure 8.

Table 2. The detailed organizational structure with roles and privileges of each employee.

Employee Name	Work Unit	Role	Privilege
Josef Noll	Dept. A	Supervisor	Administrator Final Approval Read & Write
George Kalman	Dept. A	Employee	Read & Write
Hans Christian	Dept. B	Supervisor	Administrator Final Approval Read & Write
Erik Swansson	Dept. B	Employee	Read & Write

privileges of each employee. In this section, we will see how an employee can access to the right work unit's resources based on his defined roles and relationships. In this case, the expected access situation are,

- Hans Christian and Josef Noll work as supervisor in department A and B. They hold privileges: Administrator, Read & Write and Final Approval. Therefore, they are expected to administer relevant department's administrative resources, give final approval to deliverables, and read and write ordinary documents.
- Erik Swansson and George Kalman work as employee of the department. They possess only Read & Write privilege. Hence, they should only read and write relevant department's ordinary documents.

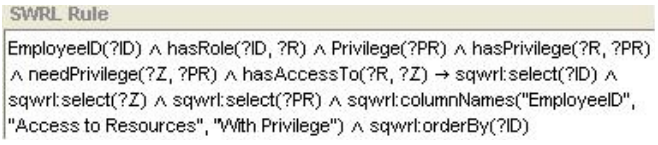


Figure 10. Rules (SWRL) along with queries (SQWRL).

A common access policy is designed according to these situations through rule using SWRL and figure 10 shows

EmployeeID	Access to Resources	With Privilege
Erik_Swansson	DocDeptB	ReadWrite
George_Kalman	DocDeptA	ReadWrite
Hans_Christian	AdminResDeptB	Admin
Hans_Christian	DocDeptB	ReadWrite
Hans_Christian	DocDeptB	ReadWrite
Hans_Christian	DeliverableDeptB	FinalApproval
Josef_Noll	AdminResDeptA	Admin
Josef_Noll	DocDeptA	ReadWrite
Josef_Noll	DocDeptA	ReadWrite
Josef_Noll	DeliverableDeptA	FinalApproval

Figure 11. Access authorization decisions derived executing rule and queries of figure 10.

the rule. Jess rule engine executed this rule and derived the access authorization decisions. The decisions are displayed in figure 11 and employees are found to get desired access to the right resources.

6 Service Interaction

This section presents a service interaction concept and demonstrates a practical implementation of a sample service interaction involving proposed authentication and authorization mechanism.

6.1 Service Interaction through Distributed Identity

A concept of service interaction using distributed identity is introduced here. Figure 12 shows graphical representation of the concept. My digital identity consists of PID, CID and SID. User personal device (e.g., SIM of mobile phone) stores PID, and CID and SID are stored mainly in the network. The later mostly are the roles and relationships of the users. Authentication providers (e.g., Bank, Government, Mobile Operators) provides the PIDs, and the roles and relationships are defined either by the user himself or by the proper authorities. The final goal is to interact the services offered by their providers through my digital identity. In this regard, users need to be authenticated and authorized. Authentication is done through PIDs and Authorization is done through role and relationships which are part of CID and SID. To facilitate the service interaction, the service providers and authentication providers should trust each other. In addition, authentication providers and users should also maintain strong trust relations. However, trust mechanism is beyond the scope of this paper. The next section presents a prototypical implementation of service access involving authentication and authorization mechanism. Authentication is done with stored binary key and authorization is realized through roles and relationships.

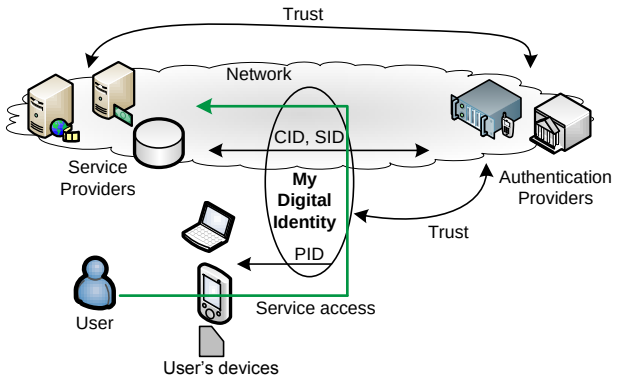


Figure 12. Concept of service interaction through distributed identity.

6.2 Prototypical Implementation of Service Access

The demonstration includes a mobile based key distribution and thereby authentication to a service by the key. It was built based on an earlier implementation of NFC-based key distribution and admittance [27]. The key generation and distribution was modified to support online access to contents of a service provider (SP). The authentication provider generates key upon request and transmit to the user through mobile phone system. The key is stored in integrated smartcard and transmitted on demand to the mobile terminal. The terminal can itself access services based on that key or, as demonstrated, can be used to perform user identification where higher security is required. In our service example, the key is transmitted over the NFC interface towards SP.

User wants to access contents remotely from a service provider using the PC. Figure 13 illustrates the steps of service access demonstration. In step 1, the access request is sent to the SP. Triggered by the request, in step 2, access control system of the provider sends a message to the authentication provider. This entity transmits access information and an access key to the mobile phone of user. We assume that Service and authentication provider belong to a common trust system and user's mobile phone number is sent to SP during service access request. The user authentication provider creates an information message (3) and a binary key (4), which is transmitted to the user's phone (here Nokia 3320) and stored in the SmartMX card of its NFC unit. The key can be transmitted over the NFC interface to NFC-enabled devices. In case, services are accessed from the PC, the key is transmitted to the PC using near field

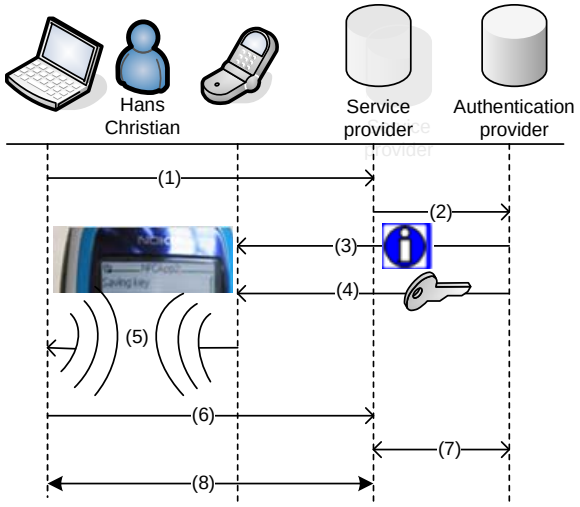


Figure 13. Prototype of online content access.

communication (5). User then presents the key to the SP (6), it then validates the key (7), and finally user gets the authenticated access (8). If services are accessed from the mobile phone, the phone number can even validate the key, provided SP has previous knowledge of key holder's phone number. Service providers also need the capability to identify the phone number from the initial service request message (1). Our implementation used Telenor's mobile network through PATS Innovation lab[10].

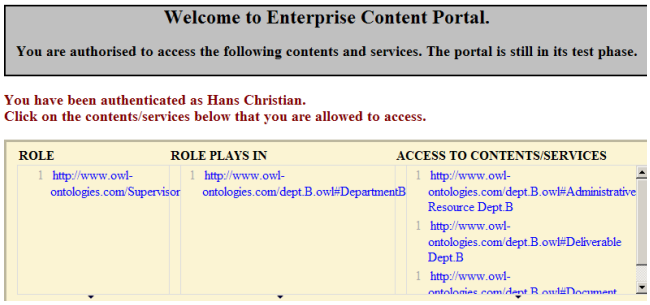


Figure 14. Prototype of access authorization to enterprise contents.

The objective is to access contents from a system of an enterprise with appropriate privileges when, each of the users has predefined roles within the organization. Hans Christian has just been authenticated by presenting a key to an enterprise content/service provider. Figure 14 illustrates a front end of the provider's content/service management system. It shows the contents, Hans Christian is authorized to access. We have seen from the discussion before that Hans is a supervisor of department B and he is permitted to access administrative resources, deliverables and documents of department B with administrator, read & write and final approval privilege. Privileges are working from the back end of the system.

[10]PATS (Program for Advanced Telecom Services), `http://www.pats.no` [retrieved on Jan. 18, 2009]

7 Related Work

This section brings in literatures and research related to secure access specifically in the area of access authorization. In this paper, we introduced user's corporate and social identities through roles and relationships. Access authorization is realized using the formal description of these roles and relationships. We are going to review some of the relevant works in this section.

Role Based Access Control (RBAC) [29] is an increasingly popular and efficient solution where users access permissions are associated with the roles, and users are made members of appropriate roles. We consider the concept of RBAC as a part of our access authorization mechanism. Besides incorporating the notion of roles, the proposed architecture includes the attribute which states 'where (department or project) one plays the roles'. Therefore, a simple notion of Attribute Based Access Control (ABAC) [36] has also been integrated in this architecture. This further restricts the relevant access authorization (toward the specific work unit) decisions. Role as a basis for authorization enables the use of constraints to support Separation of Duty (SoD). SoD is widely considered to be a fundamental precept in computer security [9],[28]. In brief, the principle states that if a sensitive task is composed of two steps, different user should do each step. We can interpret this definition in the context of the organizational environment illustrated in section 5.3. To deliver a final audit report, employee of audit department should get access to the report only to read and write specific entries but supervisor of the department has the authority to give final approval to it. In this paper, we ensure this to happen though through simple static role assignment.

In social community environment, we have chosen the relationship of an individual with the community (*member*) or with the fellow individuals (*family or friend*) as a basis to authorize access to private or community resources. Instead of relations, a concept of trust or reputation can also facilitate the access authorization [7]. In this regard, [12] introduced a distributed trust management approach to provide access to community resources. From the context of this paper, trust cannot be considered as identity traits of a person and therefore, this should not be used as the only mean to authorize access to private resources. However, the trust coupled with relationship can strengthen the privacy in

virtual community networks.

We represented the access authorization policies using OWL and SWRL. XACML (eXtensible Access Control Markup Language) is a popular access control policy language [23]. In [22], authors suggested expressing the access control policies based on OWL and SWRL citing the lack of formal semantics in XACML. KAoS [5] and Rei [21] are the two noticeable works in this regard. Policy specification language in Rei is based on OWL Lite which is less expressive compared with OWL DL. In [18], a Semantic Based Access Control Model was presented which considered semantic relations among different entities in decision making process. Therefore, use of formal representation of roles and relationships for access authorization is gaining momentum to secure a social or organization community space.

8 Discussion

This section introduces several other critical features of the proposed identity mechanism. In the 'Laws of Identity', Kim Cameron states that any sustainable and universally adopted identity architecture must only reveal the least identifying information possible with the users consent [6]. In the proposed identity mechanism, user controls how much identifying information it would reveal to the service providers by controlling the access to the primary storage of the identities (SIM card). As the services are accessed only through relevant identifiers of the PID, CID or SID, minimal disclosure of only necessary identifying information with users consent is ensured.

Sxip [2] and Windows CardSpace [3] are the two identity solutions developed by Sxip identity and Microsoft Corporation. In Sxip, membersites are typical websites that consume identity data by sending Sxip requests for user data to homesites, also websites that store user identity data. Homesites authenticate and identify users. It uses two-factor authentication solution to access services, like, online banking that requires strong authentication mechanism. Sxip 2.0 can use a third party credentials which is an interesting way to hide the use of PKI behind a software layer. Windows CardSpace uses a variety of virtual cards, each retrieving security token from the identity providers for authentication and identification to services. For greater security, user protects cards with personal identification number (PIN). To provide further assurance of secure communication, Microsoft together with other partners in industry is expected to create a new level of certificate that might contain more information than a traditional Secure Sockets Layer (SSL) certificate. These two identity solutions provide the movement of identity data over the Internet.

Storing the primary part of the identities in mobile phone provides the major advantage over the other available identity mechanisms. It is available 24 h/7 days a week, as compared to about 4 h average usage of a PC. Thus, it provides the always online functionality with availability. As, SIM card may also provide need to know authentication, some services that require minimum security can be available to the users as soon as they enter the proposed identity repository by mobile phone. Deployment of have to know authentication mechanism in SIM (ESIM) not only enhances the security to access financial services but also increases the acceptability of this identity to users. One of the useful features is portability of identifier from one device to another, especially to the devices that has no direct connectivity to 'My digital identity'. Thus, these identities can be accessed from anywhere and service continuity is possible in heterogeneous wireless environment. In case of losing or theft of SIM, we can use our PC to access 'My digital identity' which is optional and demands modification or enhancement of existing security mechanisms.

The proposed identity mechanism will create values for the users, network operators and service providers. User can use a unique identity mechanism that is simple, easy to use, digital in nature but available anywhere and portable to any device. It has the potential to replace many of the paper-based identities, such as credit card, admittance card etc. Network operators can also earn revenues by providing space for the identity repository and facilitating the additional data transfer that the system requires. As there are trust relationships among the parties involved in transactions here, the integrity and confidentiality of the transactions are ensured. Once 'My digital identity' repositories are known to the service providers, new offers can even be posted directly to these repositories.

In addition to effortless movement of identity over the Internet, the proposed mechanism supports the portability of identity data among the devices. Authentication and identification provided by the SIM card is the principle distinctive feature of it. The federated identity standards provided by the Liberty Alliance project[11] also used mobile phone identifier to access Web services. In this paper, PKI based have to know authentication mechanism has been moved to SIM card to reduce the security vulnerability. The Web-PKI suffers from insecure distribution and storage of cryptographic keys and therefore does not provide a complete chain of trust [20]. By combining the roles of CA, mobile network operators would make it easier to have a complete chain of trust around PKI because there already exists a trust relationship between mobile network operators and their customers. Gemalto, one of the leading digital security providers, is using high capacity SIM card for storing digital certificates or rights [1]. The identity repository can be used instead to store these rights that can be accessed

[11] Liberty Alliance Project, http://www.projectliberty.org/ [retrieved on Jan. 18, 2009]

through mobile phone. Thus, some overheads during data transfer can be avoided. The mechanism also ensures the portability of rights. There are many identities based on chip cards, like, memory cards and smart cards [30]. There are multiple chip cards, provided by multiple entities and single chip card, shared by few entities. If the proposed identity repository is available in the network which can be accessed anytime and from anywhere through an always online mobile phone, such various identity based chip cards might not be necessary. User needs only one smart card, ESIM card.

9 Conclusion

The paper presented a concept of digital identity, a mechanism of its management, its security infrastructures, and demonstrated a prototypical service interaction implementation. User identities are classified into personal, corporate and social identities. Part of these identities which require the highest security are going to be placed in user's personal device. The Web contains the rest of the identities. We believe the mobile phone's SIM card has the potential to be the secure personal device. User's corporate and social identities are represented through roles and relationships. Secure service access is ensured by means of authentication and authorization mechanism. Personal identifier authenticates user and authorization is achieved through user's roles and relationships. A practical implementation is demonstrated which exploits the proposed authentication and authorization mechanism. In our future work, we will focus on practical implementation of a use case that supports seamless service access in heterogeneous wireless networks using the proposed identity mechanism.

References

[1] Gemalto, a leading digital security provider. `http://www.gemalto.com` *[retrieved on Oct. 11, 2008]*.

[2] The Simple eXtensible Identity Protocol, Sxip. `http://sxip.net/` *[retrieved Oct. 11, 2008]*.

[3] Windows CardSpace. `http://cardspace.netfx3.com/` *[retrieved Oct. 11, 2008]*.

[4] T. Berners-Lee, J. A. Hendler, and O. Lassila. The semantic web. *Scientific American Magazine*, 284(5):34–43, May 2001.

[5] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Woolley. KAoS: Toward an industrial-strength open agent architecture. *In Software Agents, J. M. Bradshaw (ed.), AAAI press*, pages 375–418, 1997.

[6] K. Cameron. The Laws of Identity. `http://www.identityblog.com` *[retrieved Oct. 11, 2008]*, December 2005.

[7] H.-C. Choi, S. R. Kruk, S. Grzonkowski, K. Stankiewicz, B. Davis, and J. G. Breslin. Trust models for community-aware identity management. *In Architecture and Philosophy of the Web Identity, Reference, and the Web. IRW2006/WWW2006 Workshop, Scotland*, May 2006.

[8] M. M. R. Chowdhury, J. Noll, and J. M. Gomez. Enabling access control and privacy through ontology. *In the proceedings of 4th International Conference on Innovations in Information Technology, 2007, Innovations '07, Dubai*, pages 168–172, November 2007.

[9] D. D. Clark and D. R. Wilson. A comparision of commercial and military computer security policies. *In Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pages 184–194, April 1987.

[10] J. D. Fearon. What is identity (as we now use the word)? *Working paper,* `http://www.stanford.edu/~jfearon/papers/iden1v2.pdf` *[retrieved on Oct. 22, 2008]*, 1999.

[11] D. Fensel. *Ontologies: A silver bullet for knowledge management and electronic commerce (2nd ed.)*. Springer-Verlag, Heidelberg, 2003.

[12] T. Finin and A. Joshi. Agents, trust, and information access on the semantic web. *ACM SIGMOD, Special Issue: Special section on semantic web and data management*, 31, 5:30–35, May 2002.

[13] T. O. Group. Public key infrastructure standards. *http://archive.opengroup.org/public/tech/security/pki/index.htm [retrieved on Oct. 11, 2008]*.

[14] D. Hardt. Identity 2.0. *Presented at OSCON 2005, http://www.identity20.com/media/OSCON2005/ [[retrieved on Oct. 08, 2008]]*.

[15] M. Hogg and D. Abrams. *Social Identifications: A Social Psychology of Intergroup Relations and Group Processes*. Routledge, London, 1988.

[16] M. A. Hogg and G. M. Vaughan. *Social Psychology (3rd ed.)*. Prentice Hall, London, 2002.

[17] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member Submission*, May 2004.

[18] S. Javanmardi, M. Amini, R. Jalili, and Y. Ganjisaffari. SBAC: Semantic Based Access Control. *In the proceedings of the 11th Nordic Workshop on Security IT Systems, Linkoping, Sweden*, pages 157–168, October 2006.

[19] R. Jenkins. *Social Identity*. Routledge, London, 1996.

[20] A. Jsang and G. Sanderud. Security in mobile communications: Challenges and opportunities. *In the proceedings of the Australasian Information Security Workshop, Adelaide, Australia*, February 2003.

[21] L. Kagal. Rei:A Policy Language for the Me-Centric Project. *TechReport, HP Labs*, September 2002.

[22] H. Li, X. Zhang, H. Wu, and Y. Qu. Design and Application of Rule Based Access Control Policies. *In proceedings of the International Semantic Web Conference Workshop on Semantic Web and Policy*, pages 34–41, 2005.

[23] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shah. First experience using XACL for access control in distributed systems. *In proceedings of ACM Workshop on XML Security, VA, USA*, October 2003.

[24] G. J. McCall and R. Simmons. *Identities and Interactions*. NewYork: Free Press.., 1966.

[25] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. *Proceedings of International Semantic Web Conference*, pages 549–563, 2004.

[26] J. Noll. Services and applications in future wireless networks. *In Telektronikk, Q4/2006*.

[27] J. Noll, J. C. L. Calvet, and K. Myksvoll. Admittance Services through Mobile Phone Short Messages. *Proceedings of the International Multi-Conference on Computing in the Global Information Technology (ICCGI 2006)*, pages 77–82, July 2006.

[28] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *In Proceedings of the IEEE*, 63, 9:1278–1308, September 1975.

[29] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29, 2:38–47, February 1996.

[30] S. Sengodan. On secure mobile identity provisioning. *In the proceedings of Wireless World Research Forum Meeting 15, Paris, France*, December 2005.

[31] M. K. Smith, C. Welty, and D. L. McGuinness. OWL Web Ontology Language guide. *W3C Recommendation*, February 2004.

[32] S. Stryker. *Symbolic Interactionism*. Menlo Park, CA: Benjamin/Cummings., 1980.

[33] H. Tajfel and J. Turner. An integrative theory of intergroup conflict. *In William G. Austin, Stephen Worchel (ed.), The Social Psychology of Intergroup Relations. Monterey, CA: Brooks-Cole*, pages 94–109, 1979.

[34] R. H. Turner. The Role and the Person. *American Journal of Sociology*, 84:1–23, 1978.

[35] White Paper. High capacity SIMs. *http://visionmobile.com/whitepapers.html [retrieved on Oct. 11, 2008]*, March 2006.

[36] E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. *In the proceedings of the IEEE International Conference on Web Services (ICWS05), Orlando, Florida*, pages 561–569, 2005.

AMISEC: Leveraging redundancy and adaptability to secure AmI applications

José M. Moya, Juan Carlos Vallejo, Pedro Malagón,
Álvaro Araujo, Juan-Mariano de Goyeneche, Octavio Nieto-Taladriz
Universidad Politécnica de Madrid
ETSI de Telecomunicación
Ciudad Universitaria, s/n, 28040 Madrid, Spain
{josem,jcvallejo,malagon,araujo,goyeneche,nieto}@die.upm.es

Abstract

Security in Ambient Intelligence (AmI) poses too many challenges due to the inherently insecure nature of wireless sensor nodes. However, there are two characteristics of these environments that can be used effectively to prevent, detect, and confine attacks: redundancy and continuous adaptation. In this article we propose a global strategy and a system architecture to cope with security issues in AmI applications at different levels. Unlike in previous approaches, we assume an individual wireless node is vulnerable.

We present an agent-based architecture with supporting services that is proven to be adequate to detect and confine common attacks. Decisions at different levels are supported by a trust-based framework with good and bad reputation feedback while maintaining resistance to bad-mouthing attacks. We also propose a set of services that can be used to handle identification, authentication, and authorization in intelligent ambients.

The resulting approach takes into account practical issues, such as resource limitation, bandwidth optimization, and scalability.

Keywords: *Ambient intelligence, reputation system, security framework for wireless sensor networks.*

1. Introduction

In essence, an intelligent environment is a distributed system that collects data from a wireless sensor network, processes this data, and enriches the environment with new meaning. These semantic enhancements can be used by other applications running on top of our system to make decisions.

Security concerns are key issues in ambient intelligence (AmI) since its earliest inception (Weiser, 1993). Many researchers clearly recognize the inherent challenge that an invisible, intuitive and pervasive system of networked computers holds for current social norms and values concerning privacy and surveillance. In fact, the increasing attack rate has become the bottleneck of adopting next-generation services and applications. A study from the Computer Security Institute reveals that a random sample of 223 organizations had lost hundreds of millions of dollars in 2002 due to security attacks [1].

For example, Brumley and Boneh [7] developed a timing attack for the OpenSSL implementation of RSA in a real TCP/IP network. This low-cost attack exploits some asymmetries introduced by two optimizations used in the OpenSSL implementation. Even in OpenSSL, that is considered to be quite reliable and secure, and it is used in many servers around the world, it is possible to find asymmetries that reveal some data of the cryptographic keys. And these asymmetries can be used to implement a real attack. Using OpenSSL or something equivalent for sensor communications would be impractical in most cases, and therefore the security threats become much worse as many more attack opportunities arise.

Three factors contribute to make security in intelligent environments a very difficult problem: 1) many nodes in the network have very limited resources; 2) pervasiveness implies that some nodes will be in non-controlled areas and are accessible to potential intruders; 3) all these computers are globally interconnected, allowing attacks to be propagated step by step from the more resource-constrained devices to the more secure servers with lots of private data.

Usually, security issues are addressed, in a similar way to services in a network of general-purpose computers, by adding an authentication system and encrypted communications. First, the resource limitations make the embedded computers especially vulnerable to common attacks.

In previous work [19], we demostrated that current ciphers and countermeasures often imply more resources

(more computation requirements, more power consumption, specific integrated circuits with careful physical design, etc.), but usually this is not affordable for this kind of applications. But even if we impose strong requirements for any individual node to be connected to our network, it is virtually impossible to update hardware and software whenever a security flaw is found. It has already been stressed the need to consider security as a new dimension during the whole design process of embedded systems [17, 23], and there are some initial efforts towards design methodologies to support security [2, 5, 24], but to the best of our knowledge no attempt has been made to exploit the special characteristics of AmI environments.

AmI applications have to live with the fact that privacy and integrity can not be preserved in every node of the network. This poses restrictions on the information a single node can manage, and also in the way the applications are designed and distributed in the network.

Of course, the inherent insecurity of embedded systems should not lead us to not try hard to avoid compromises. We should guarantee that a massive attack can not be fast enough to avoid the detection and recovery measures to be effective. Therefore we should design the nodes as secure as the available resources allow.

In spite of the disadvantages of AmI environments from the security point of view, they provide two advantages for fighting against attacks:

- Redundancy. AmI environments usually have a high degree of spatial redundancy (many sensors that should provide coherent data), and temporal redundancy (habits, periodic behaviors, causal dependencies), and both can be used to detect and isolate faulty or compromised nodes in a very effective manner.

- Continuous adaptation. AmI environments are evolving continuously, there are continuous changes of functional requirements (data requests, service requests, user commands...), nodes appear and disappear continuously and therefore routing schemes change, low batteries force some functionality to be migrated to other nodes, etc.

In this article we propose a more secure approach to the design of AmI applications by exploiting these two properties. Section 2 describes our approach in detail. In section 3 we review some relevant attacks, the countermeasures that have been proposed previously, the requirements that these threats impose to our design strategy and demonstrates how this approach can detect and confine them. In section 4, we draw some conclusions.

2. AMISEC architecture

2.1 Overview

We focus on the development of secure applications in future wireless sensor networks, where many sensors provide data about observable magnitudes from the environment, and many actuators let the system act on the state of the environment.

Following the Ackoff taxonomy for the content of the human mind, we classify the content of the "ambient mind" into four categories:

1. Data: Symbols. It simply exists and has no significance beyond its existence (in and of itself).

2. Information: Data that is processed to be useful; provides answers to "who", "what", "where", and "when" questions.

3. Knowledge: Application of data and information; answers "how" questions.

4. Intelligence[1]: Appreciation of "why". It is the process by which new knowledge is synthesized from the previously held knowledge.

The main characteristic of an intelligent ambient is the semantic enrichment of environment based on the processing of data obtained from the environment using a sensor network. This "ambient mind" enhances the semantics of the environment by adding meaning to the objects. The objects are conscious of the "who", "what", "where", "when", "how", and "why".

Data is obtained by sensor nodes, but as they are not trusted, most of the remaining processing should be done in secure servers so that confidentiality attacks do not succeed (note that data has no meaning by itself). Data is sent to servers where it is processed to generate information, and then knowledge, and then understanding, and then new meaning, which is returned back to the environment. Individual nodes may be insecure, but the system should always continue its function of semantic enhancement. Moreover, attacks of individual nodes should not affect the decisions based on data from the environment. These requirements are achieved by perusing redundancy to discard data from the compromised nodes, and by changing the network structure and behavior at a speed that is fast enough to prevent a chained attack to spread.

Figure 1 shows the data flow in the environment. As confidentiality attacks become more dangerous as data is further processed, there should be little or no processing at all in the sensor nodes, which are more vulnerable.

[1] Actually, this category comprises two from the Ackoff taxonomy: understanding and wisdom.

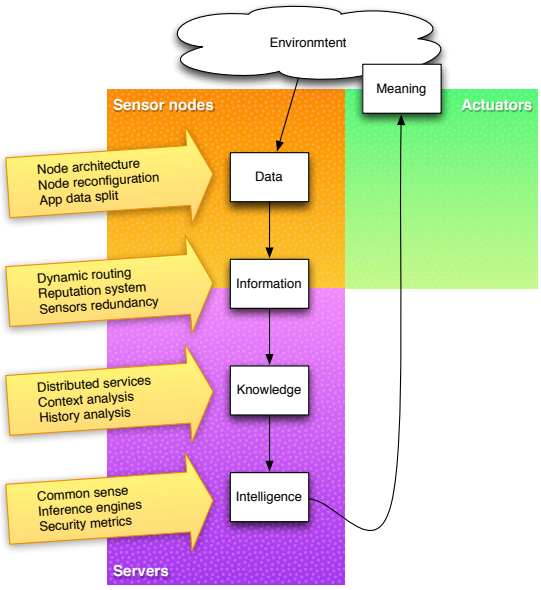


Figure 1. Overview of the data processing flow in the AMISEC environment and the security measures.

2.2 Network model

We consider the network composed by two kinds of nodes: wireless nodes and servers.

- Wireless nodes. They provide data to the network to enable decisions to be made. In our model, decisions are made primarily in secure servers, and therefore the main task of these wireless nodes is sending data to the servers. The more data is sent to the servers, the more redundancy can be used to discard bad data and to detect failures or intrusions. But also, the more data is sent, the more bandwidth is used and the more energy is consumed, so we have to reach a compromise. There are many wireless nodes in an intelligent ambient, so they have to be inexpensive, what usually means very limited resources, battery-powered, not maintained and hence insecure; an intruder may have physical access to them.

- Servers. They receive data from sensors and make decisions in order to reach the applications objectives. These decisions may imply to act in the environment and therefore they have to be secure. Servers are usually well maintained, wire-connected and their resources are not usually constrained at all.

In order to improve network scalability and throughput, we use a clustering technique based on Random Competition based Clustering (RCC) [29] to construct a multi-level network structures. Previous approaches [3, 4, 18] group nodes into clusters, and within each cluster a node is elected as a cluster head. Cluster heads together form a higher-level network, upon which clustering can again be applied. This structure simplifies communication and makes it possible to restrict bandwidth-consuming network attacks like flooding to a single cluster.

For a wireless network with n nodes capable of transmitting at $Wbits/s$, according to [14], the throughput, T, for each node under optimal conditions is

$$T = \Theta\left(\frac{W}{\sqrt{n}}\right)$$

Thanks to the clustering approach, in a two-level mobile backbone network where the number of nodes is n and the number of clusters is m, the throughput in the lower level becomes

$$T_1 = \Theta\left(\frac{W_1}{\sqrt{n/m}}\right)$$

and in the higher level

$$T_2 = \Theta\left(\frac{W_2}{\sqrt{m}}\right)$$

Node clustering, however, reduces redundancy and introduces single points of failure, as an intruder could control a whole zone by attacking its cluster head. The solution we propose is to introduce redundancy again. Every node in the network will have several cluster heads and will distribute messages randomly between them. This additional redundancy does not reduce the maximum throughput because at any given time the network structure is exactly the same as in the pure RCC scheme.

It may be argued that for every node to have two cluster heads, we need to double the backbone nodes so that there are twice as much backbone nodes in the coverage area. While it is true that more nodes have to belong to the backbone, this does not imply any reduction of the attainable throughput, as at any given time half the backbone nodes will not be used as such, and therefore the network structure remains exactly the same as in the pure RCC case. On the contrary, the burden of routing backbone messages is more distributed and therefore the penalty in energy consumption of being a cluster head is significantly reduced.

2.3 Assumptions

We assume that servers are secure and reliable.

The number of wireless nodes is assumed to be huge compared to the number of servers.

Due to being physically accessible and resource-constrained, wireless nodes are considered to be vulnerable. We assume an intruder can seize control of any wireless node in a minimum time ta.

There is a working service location system in the network, and it is secure and reliable. This article will not address the problems of deployment and operation of this service. We assume that every node in the network knows how to reach any particular service.

As redundancy is good to detect and isolate attacks, any device providing useful information should be welcomed. Therefore, we assume that new wireless nodes can be added dynamically to our network without any restriction. Our architecture should assure that a continuous addition of bad nodes will not affect to the global behavior.

2.4 System architecture

The AMISEC approach to the previously described threats is based on leveraging the two weapons that we have to detect and resist to attacks and failures: redundancy (spatial and temporal), and continuous adaptation. Also, we know that individual wireless nodes are vulnerable to attacks, and therefore no important decision should be made by a single node and no significant information should be stored in a single node.

We propose a software architecture based on many independent agents with simple and clear responsibilities. The term agent is heavily overloaded and should be defined more precisely. An AMISEC agent is an independent piece of software that is able to act on your behalf while you are doing other things (they are proactive), and it does this based on its knowledge of your preferences and the context. This knowledge is stored in servers and it is available to the network nodes through the use of passive services.

Figure 2 shows the main AMISEC components. As can be seen, there is no direct communication between sensors and actuators, in order to avoid an intruder to modify the state of the environment while not preventing the free addition of sensor nodes to the system.

Individual sensor nodes are not trusted by default, and therefore the notion of trust is built dynamically by comparing a sensor with its neighbourhood. For this reason, every agent that needs to take into account data coming from sensor nodes or any derived information uses a trust-based decision framework that is further described below.

2.4.1 Trust-based decision framework

We follow the definitions and beliefs of Boukerch et al. in [6] concerning the distinction between trust and reputation.

Trust is the degree of belief about the future behavior of other entities. Trust is subjective and it is based on past experiences.

Reputation, on the other hand, is the global perception of an entity's behavior, and it is based on the trust that others hold on that entity. It is mostly objective and it has some influence in the evolution of trust in every node.

To consider a data item to be valid we use two consistency tests. The data item is said to be s-consistent or consistent with the spatial redundancy if it is consistent with the data provided by the majority of sensors that provide measurements of the same variable. For example, for a presence event from a PIR detector to be valid, the majority of nodes monitoring the same area should also detect presence. In this evaluation every sensor is weighted with the trust value the receiving node has about the source node.

A second way to discard bad data is to evaluate each data item against temporal data redundancy. Each routing element stores a limited set of previous values for each variable directly routed through itself. The data item is said to be t-consistent if the variation against previous data is normal for that variable. For example, if a temperature value changes drastically and it is not maintained during some time, maybe a routing element has been attacked.

Both properties, s-consistency and t-consistency, are dependent on the variable being measured.

To model trust and reputation in our agent system, every node in the network maintains a trust table with entries for every relevant neighbor node.

When a new node is discovered, the initial trust value is 0.

Whenever a new message containing a new measurement of the external variable v arrives, trust on node i is recalculated as follows.

$$d_v(t) = A_v(\{\tau_i(t-1), d_{vi}(t)\})$$
$$\tau_i(t) = T(\tau_i(t-1), d_v(t), \overline{H_{vi}})$$

$\overline{H_{vi}}$ represents all the data values of the variable v provided by node i that are stored in this node (history is usually truncated to reduce memory requirements). A_v is an aggregation function that depends on the variable being measured, and it does not take into account data coming from a node with negative or zero trust value. T is also an aggregation function with these properties:

- If $\tau_i(t-1)$ is negative, the data item is discarded and no further processing is done for this message (repeated inconsistencies may lead to negative values of trust).
- If the new data element $d_{vi}(t)$ is s-inconsistent and t-inconsistent, it is stored in the local history (discarding the oldest value), but it is not taken into account for trust recalculation.

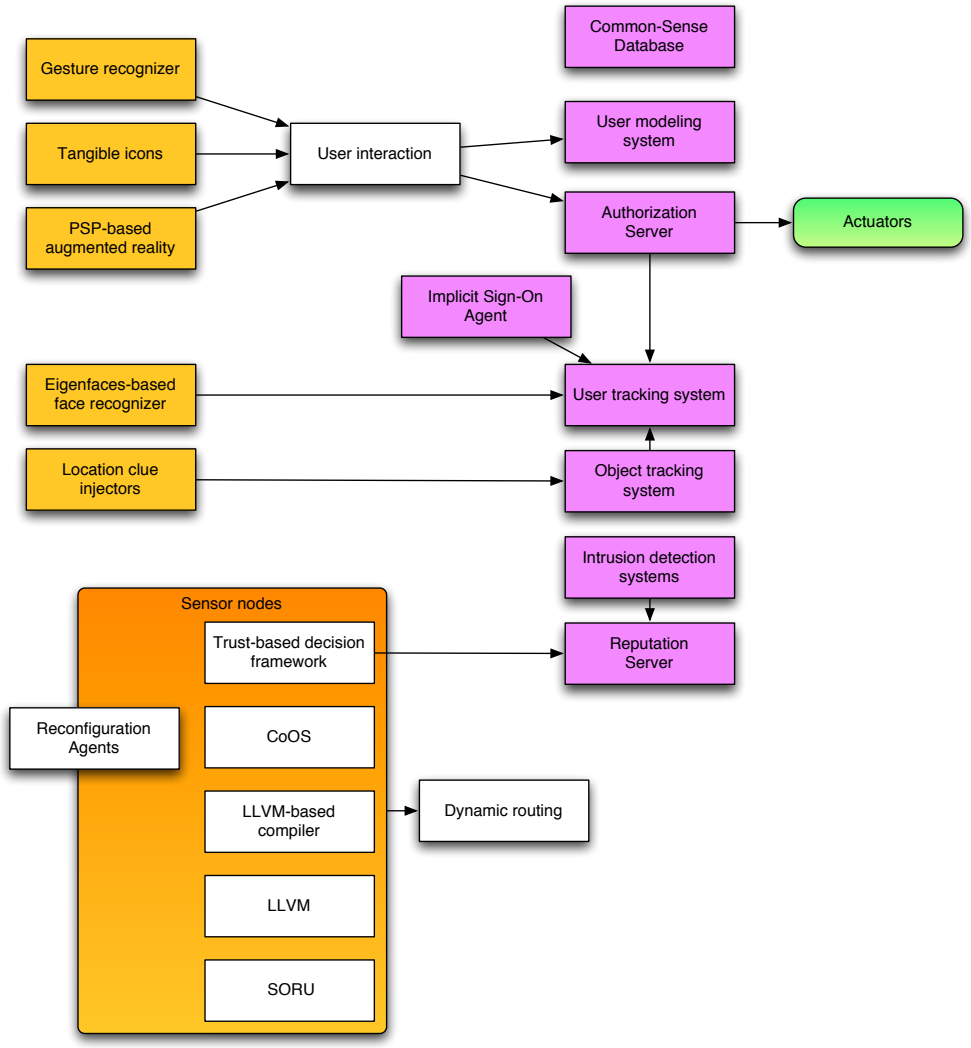


Figure 2. Main AMISEC components.

- If it is s-inconsistent with other sensors' data but t-consistent with previous values of the same sensor, trust on sensor i decreases.

- If it is s-consistent and t-consistent and current trust is positive, trust increases.

As can be seen, trust computation condenses historical information, and therefore it is bad, as we lose redundancy. On the other hand, resources are tightly constrained and we have to reduce storage requirements to a minimum.

To avoid some attacks, temporal disappearance means loss of positive trust (not negative). Whenever it appears again, it will get a 0 trust value.

There is a second method to feed trust values back from redundancy analysis: reputation messages from the servers.

From time to time, nodes communicate their trust tables to the servers. This is done at the routing level by adding this trust information to messages that are being sent to the same destination. Servers are not resource constrained by assumption, and therefore they can store all the historical information for future analysis. The adequate combination of all the trust data of a zone generates the global reputation data:

$$\rho_{vi}(t) = R(\rho_{vi}(t-1), H_{vi})$$

Where H_{vi} represents all the history of data values of the variable v provided by sensor i, and R is another aggregation function. Well-behaved nodes increase their reputation; bad-behaved ones decrease their reputation. Multiple agents can be running on the trust servers to look for

attack evidences in the message history, and proactively reduce reputation values of suspect nodes.

Whenever a server decides that it has to act in the environment by modifying trust values for ill-behaved nodes, it broadcasts the reputation information of all the nodes in that zone. This message is repeated from time to time until the data the server receives from that zone is consistent with the global reputation information.

A wireless node will never take into account this reputation information unless it has been received from different routers (cluster heads). Thus, redundancy in routing paths and trust merging in secure servers allows us to feed good and bad reputation back to the network without being vulnerable to bad mouthing attacks.

The trust data sent to the servers is enough to detect most, if not all, common attacks. However, it is not enough to find the concrete faulty or compromised node, and therefore the servers would not be able to confine the attack. The solution we propose is to include the routing path in some of the messages. This way, by analyzing the paths of messages with t-consistent and s-consistent data it is easy to discard well-behaved nodes. Note that routing paths coming from a compromised node could have been faked.

The confinement agents act directly by decreasing the reputation values of the suspect nodes.

Parameter	Description
Redundancy-related	
N_p	Number of reputation tables stored in a node.
N_d	Number of values stored for each sensor/value pair.
N_r	Number of routers per node.
Adaptation-related	
t_τ	Time between trust data messages sent to the reputation servers.
t_ρ	Time between reputation data messages from the servers to the nodes.
t_v	Time between sensor data messages from the sensor nodes to the network.
t_r	Minimum time between messages containing route information.

Table 1. Parameters that can be adjusted dynamically to adapt the environment to possible attacks.

A number of parameters (see table 1) can be dynamically adjusted in order to adapt the environment to possible attacks. If the risk increases, we increase the local amount of redundancy around the affected area.

2.4.2 Sensor agents

Sensor agents are the simplest ones. They usually run on wireless nodes and provide measured data of external variables to the network, by sending messages to their routing agents. The message rate depends on the variation rate of the variable being monitored. This message rate should be enough to ensure that data items do not change too fast and therefore temporal redundancy can be used to detect failures or attacks.

Each sensor agent is associated to a sensor device and generates a sequence of measurements $d_{vi}(t), d_{vi}(t+1), ...$ where v is the variable being measured and i is the sensor agent id. Each data item is annotated with a time stamp, to detect temporal anomalies.

As previously stated, there is not a single routing agent for each sensor agent, and this agent decides randomly what routing agent to use for every message.

Although they do not consume data from other sensors, they need to maintain a trust table for their routing elements, that will only evolve with reputation information coming from the servers. Unlike in routing elements, the initial trust value for a routing element is positive, and the distribution of messages is uniform between all the routing nodes with positive trust.

2.4.3 Actuator agents

Actuator agents operate physically on the environment (light switches, electronic equipment controls, alarms, etc.). They are especially critical because 1) they are usually not redundant, and 2) any operation on them causes a physical effect on the environment. Therefore the nodes running actuator agents should be at least as tamper-resistant as the physical element they control. To ensure that an intruder can not operate remotely on an actuator, only servers can send operation requests to these agents and they should use robust asymmetric encryption algorithms. As security and processing requirements are higher, these nodes are usually main powered.

The data flow goes from sensors to servers and from servers to actuators. There is no feedback from actuators to servers. So if an actuator is attacked, the assailant will not be able to access to the others entities in the network.

Logically, an actuator works as a passive service, but it also develops a trust model of its environment, which is fed to the servers.

2.4.4 Aggregation agents

Aggregation agents reduce the redundancy by combining several data items using a known aggregation function. The only reason to apply these aggregations is to reduce the

amount of data sent to the servers, allowing the system to scale.

Trust computation implies also an aggregation of spatial and temporal redundant data that is held in a node.

2.4.5 Services

Services are passive elements that can be used by other nodes in the network. They usually run in servers.

Some of the services that have important roles for security reasons are: object tracking system, user tracking system, user modeling system, and common sense database.

2.5 Identity and authentication

In this kind of environments there are two types of identity: object identity and user identity.

Objects are every traceable element in the network (a wireless sensor node, a camera, a remote control device, etc.). They are freely added to the network and they will only be isolated by the system in case of bad behavior.

Object identity is handled by the object tracking system, a server that stores and processes all the location information of network objects.

Different agents provide location information about the objects in the network.

From the security point of view, the main purpose of the object tracking system is to be able to detect and confine sybil attacks [11].

Authentication is implicit and linked to the concept of reputation. When the system has enough consistent data from an object identity, its reputation will grow and it is considered to be authenticated.

User identity is handled by the user tracking system. User identities are logical identifiers that are used to handle permissions in the environment. They are linked to objects automatically, based on the analysis of the data coming from the environment, the user model (preferences, habits, etc), and the common sense (we use a common sense database based on OpenMind's).

2.6 Authorization service

As previously seen, actuators have to be more secure because they can operate on the environment. No agent is allowed to use directly an actuator. They send an actuation request to the authorization service, and this service, if the object is linked to a user identity with permissions and the action is considered to be secure, will use the actuator. In our current implementation the authorization service holds the public keys for every actuator in the system and every operation message is encrypted with the public key of the actuator.

2.7 Surveillance agents

Common intrusion detection systems, as well as more specific analyses can be run in the servers to detect intrusions or failures. These systems can confine a detected intrusion by changing reputation tables and identity information.

2.8 Application-level issues

Information handled by a single wireless node can not be significant. This poses restrictions on the way the applications are designed.

3. Evaluation

Nodes of a sensor network need to access, store, manipulate and communicate information. In AmI, nodes make decisions based on received data. Therefore, the system must guarantee data reliability. Some applications will require the use of sensitive information. In that case, measures to ensure data confidentiality should be taken into account. In this section, we will analyze the different kinds of attack that a sensor network is exposed to.

The next sections classify the different threats attending to their primary focus.

3.1 Confidentiality attacks

Confidentiality attacks attempt to access to the information stored in the sensor network. They can be further classified attending to the target of the attack:

- Attacks to the confidentiality of communications.
- Attacks to the confidentiality of node information (data generated in the sensor waiting to be sent to a server, service information stored in the network, and server information).

In a closed system with high-resources devices, information can be protected using cipher algorithm and physical access control. However, sensor networks are more vulnerable due to their characteristics:

1. Nodes have very limited resources.
2. Potential intruders may physically access to them.
3. Wireless communications.

The network can use well-suited cipher algorithms [20] to provide security against attacks to communications. Due

to conditions 1 and 2, nodes are more vulnerable to the attacks than communications. Some approaches suggest ciphering stored data [25]. Nevertheless, a combination of logical (cryptography weakness and Trojan horses), and physical (DPA, SPA, micro-probing, reverse engineering) attacks could break the ciphering and access the information.

Due to the characteristics of the sensor nodes, it is not possible to secure its data against attacks. Even if we cipher the information in the devices, an attacker could use an approach based on logical and physical attacks that could break the ciphering. Since attackers have physical access to the nodes and nodes have limited resources, confidentiality should be based in the main characteristics of sensor networks: distribution and redundancy.

3.1.1 Attack to the confidentiality of node information

Sensor agents. In this kind of attack, the intruder accesses to the information stored in a sensor. If the attack successes, the attacker will obtain the information stored in it, but it is only raw data, not significant by itself. In addition, mapping that information with a concrete user is impossible because mapping information is stored in servers or distributed among a very large number of nodes. While the number of nodes holding some particular information remains much higher than the number of attacked nodes, attackers will not be able to obtain meaningful information.

Actuator agents. These agents do not store other information than the status of the physical device they control and the trust table for its routers.

Aggregation agents. By attacking an aggregation agent or a node that runs an aggregation agent, an intruder may gain access to redundant local raw data, but anything else. Redundant data is useful to discard bad data, but it gives no extra information.

Decision-making agents. They run in servers, which are not physically accessible, and have enough resources to keep the information secure.

3.1.2 Attack to the confidentiality of communications

In this attack, an intruder listens to the channel trying to obtain some information. Due to sensor redundancy and information distribution, the attacker should break all communications between sensors and routers to obtain some significant information. The use of some ciphering algorithms will help protecting the system. Since the network is big enough, an attacker that listens to the channel will obtain only a set of $d_{vi}(t)$. By definition, that set will not represent any meaningful information, so the attack will fail.

3.2 Denial of service attacks

A Denial of Service (DoS) attack is an attempt to interrupt, disrupt, or destroy services and operations in a system, which includes:

- *Jamming, collision and flooding*: These attacks consist in interfering in communication by sending messages through several protocol layers. The immediate effect of these attacks is the loss of part of the messages from the nodes of the affected area. The affected area depends on the layer in which it occurs. The upper the attack occurs on the protocol stack, the more it spreads. So the scope of these attacks could be zone or global depending on their dimension and the layer where they occur (physical, link, or transport layers). Wood and Stankovic [28] explain several countermeasures for these attacks: they suggest confinement, small frames, error-correcting codes and client puzzles.

- *Neglect and greed*: This simple form of DoS attack focus on a router vulnerability by arbitrarily ignoring all or some messages. It is especially dangerous in environments using hierarchical routes and static routing protocols. A possible solution would be a routing protocol with several paths available [28].

- *Misdirection, blackholes and wormholes*: These attacks are very difficult to avoid, detect and confine. Authorization and monitoring have been proposed to avoid them. However, it is not possible to deploy a secure wireless sensor networks based exclusively on ciphering and authorization. It is necessary to supply additional techniques to reinforce the system. We will use redundancy again to detect these attacks. There exists some countermeasures consisting on enhanced protocols like [9], however they require too many resources to be used in tiny nodes.

Now, we will show how our system can detect and confine the denial of service attacks.

3.2.1 Jamming, collision and flooding

Whether it is jamming, collision or flooding, the effects in the network are similar: loss of messages and node disappearance. The seriousness and extension of the attack depends on the number of nodes, the stack layer where it takes place and several other parameters. Nevertheless, it leads some nodes to disappear. As no new value from these nodes arrives to the routers, as trust tables are sent to the servers, the global trust service will soon discover that the latest values coming from these nodes are obsolete and it will mark them as lost.

The detection of the attack can be performed when a group of nodes in the same area disappears suddenly. If a node with positive reputation disappears temporally its reputation will be decreased. This measure will also affect directly to the routers in the area. Therefore, a message will not be sent through an affected router, avoiding the zone.

Flooding attack could be more dangerous if messages are scattered and the whole network is affected. But if the reputation of a faked node is decreased, its forwarded messages will not be routed and, therefore, harm will not spread.

3.2.2 Neglect and greed, and blackholes

A router may neglect to route all or some messages, but every node has two or more routers that are used randomly, and so eventually the messages will arrive to the destination.

Some of the messages include their own route, and the servers analyze the routes of consistent messages to find out the routers which do not route properly. A feedback of negative reputation for these routers will cause messages to follow other routes avoiding these malicious routers.

3.2.3 Misdirection and wormholes

Local attacks can get worse if the compromised node stops routing properly, changes the values notified by some sensors, or teleports messages to other area of the network.

A combined use of localization information (object tracking system), and route analysis for messages coming from the same area (redundancy in routing elements will ensure that not every message will go through the wormhole), allows to discover easily the bad routers. There are some proposals similar to this one, like in [15, 27] where authors propose a method based on location information of each node join to identity information in messages or like in [26] where a statistical process of network data is used to detect wormholes. AMISEC manages the required data so both are feasible solutions for our system.

Again, once the malicious routers have been detected, it is possible to confine the compromised nodes by decreasing their reputation. If a router has a low reputation it will be probably not chosen for routing messages. And redundancy in routing elements ensures that the new reputation table will eventually arrive to any node in the network.

Trust tables going from the sensor nodes to the servers and reputation tables coming back from the servers can also be altered by a compromised node, but redundancy again allows discarding bad messages.

3.3 Integrity attacks

Integrity attacks try to alter the normal behavior of the system by modifying the data stored in nodes. Although DoS attacks can be considered as integrity attacks as service interruption is one kind of bad behavior, we prefer to treat them separately because here the focus is on the data, instead of the communications.

3.3.1 Tampering and homing

These attacks are very difficult to avoid due to the weakness of wireless nodes. But these are clear cases of local attacks. Local or node attacks are not relevant for the AMISEC model, since redundancy allows losing nodes without any impact in the behavior. Negative reputation can be used from the servers in order to confine these attacks.

Even if integrity of individual nodes is difficult to achieve, the use of redundancy can reduce or eliminate the impact on the global system.

3.4 Identity attacks

Malicious nodes can pretend to be other nodes in order to implement one of the attacks mentioned above. We will consider four different types: clone, thief, mole and sybil.

- The *clone attack* consists in duplicating an operating node. Both nodes, simultaneously, communicate with the same identity.
- In the *thief attack*, a malicious node steals an operating node its identity and replaces it in the network. The malicious node stops original node's operation and takes advantage of its reputation and trust levels.
- A *mole* is a malicious node that behaves as a well-operating node, with a fabricated identification, to achieve high levels of trust and reputation. Once inside, it can attack the system from a privileged position.
- The *Sybil attack* occurs when a malicious device presents multiple identities, as if it were multiple nodes, in order to control a substantial fraction of the system. This attack reduces the effect of the system's redundancy without the need of numerous physical nodes. The attacks can be performed at any layer of the protocol stack, but they are more profitable in the upper layers, like network or application.

The first three attacks are carried out by individual malicious nodes, and they can be considered special cases of the Sybil attack. The Sybil attack was first introduced in [11]. Newsome [22], Karlof [16] and Zhang [30] make thorough descriptions of the taxonomy, threats and countermeasures of identity attacks, focusing on the Sybil attack. We can find three main types of solutions to the identity attacks: resource testing, cryptography and location-based.

Resource testing solutions assume that devices are limited in some resource [11]. The solutions consist in testing a limited resource and checking that each identity has no less capability than a physical node. The resource tested in wireless sensor networks, according to Newsome [22], is the radio communication capability, considering that a device can access only to one radio channel at a time. Each identity has a channel assigned and they must send a message through it simultaneously. The system detects an identity of a Sybil attack when it receives no message in its channel. Accurate synchronization between the monitoring devices is needed and, if we have more identities than channels, we can't perform the test to every identity at the same time, so the detection rate decreases.

Cryptography schemes base their efficiency in secure communications, and the different solutions differ in how to establish the keys: the key agreement process. They can have a key server with the public key of all nodes, and only establish a key through the key server. Another scheme uses the self-enforcing scheme approach, based on asymmetric cryptography with public key. Efficient implementations of ECC [] can be used in sensor networks to establish secure links, but it is not enough to avoid the Sybil attack, because a malicious device may have more resources than the normal nodes. The third key agreement mechanism is key predistribution scheme [8, 12, 13]. In these systems each sensor has a subset of the system keys and a secure link is established between nodes which have at least one key in common. If a node is compromised, several keys are known by the malicious device. If more nodes are compromised, the attackers can obtain a substantial fraction of the system keys.

Location based solutions [10, 21], check that no identities are at the same position. The solutions assume that the sensor nodes are static, but real AmI applications have heterogeneous networks, with static and moving nodes. The accuracy of the location system should be high due to the high density of sensors inherent to AmI applications.

Clone, thief and mole attacks use only one identity, so their effect is the same as compromising one sole node. It is proven, as shown in previous sections, that the system adapts to individual attacks. If the node's behavior is consistent with the other nodes, the attack is undetectable, but the information obtained is not significant. In the clone attack the system can detect that the same identity is being used in two different locations, so the server would reduce the reputation of both nodes.

On the other hand, the Sybil attack can be dangerous to the system because it reduces the effect of the system's redundancy. Our architecture solves the Sybil attack problem by reducing its attack rate. When an aggregation agent receives information from an unknown node, the trust level default value is zero. This is enough to send data from this node to the servers to collect behavior history, but not enough to be taken into account in any decision or aggregation. If the node behaves correctly, its reputation will grow eventually, but always at a controlled rate. If many sensors are appearing in a short time in the same area, the required time to have positive reputation will increase.

4. Conclusion

Wireless Personal Area Networks are based on many wireless, low cost, low power, and low resources nodes. These characteristics and the possibility to access physically to the node make the nodes highly vulnerable to attacks. Cryptography appears as clearly insufficient to maintain data confidentiality and integrity in the network.

We have proposed a holistic solution that assumes this node vulnerability to address security issues in an intelligent ambient based on massive wireless sensor networks.

Redundancy and fast continuous adaptation have been identified as the key weapons to defend the system against attacks, and they are used consistently to cope with security issues at different levels.

The AMISEC architecture is based on an agent system with supporting services. Data flows from the sensors to the servers, where it is processed returning relevant semantic enhancements back to the environment. Agents running in insecure wireless nodes never hold a significant information unit, what preserves global confidentiality, and decisions are made in servers, what preserves integrity if redundancy is used adequately.

Most attacks are detected by the analysis of the redundant data available in the network and collected in the servers.

Decisions at different levels are supported by a trust-based framework where trust data only flows from the sensors to the servers and reputation only from the servers to the sensors.

The resulting approach takes into account practical issues, such as resource limitation, bandwidth optimization, and scalability.

Based on these results we claim that our approach provides a practical solution for developing secure AmI applications.

Acknowledgements

This work was funded partly by the Spanish Ministry of Industry, Tourism and Trade, under the CENIT Project Segur@, and partly by DGUI de la Comunidad Autónoma de Madrid and Universidad Politécnica de Madrid under Grant CCG07-UPM/TIC-1742.

References

[1] Computer crime and security survey. Computer Security Institute, 2002.

[2] D. Arora, A. Raghunathan, S. R. M. Sankaradass, N. K. Jha, and S. T. Chakradhar. Software architecture exploration for high-performance security processing on a multiprocessor mobile soc. In *Proceedings of the 43rd Annual Conference on Design Automation*, pages 496–501, San Francisco, CA, USA, July 24 - 28 2006.

[3] S. Bannerjee and S. Khuller. A clustering scheme for hierarchical control in wireless networks. In *In Proceedings of IEEE INFOCOM*, 2001.

[4] S. Basagni. Distributed clustering for ad hoc networks. *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 310–315, June 1999.

[5] L. Benini, A. Macii, E. Macii, E. Omerbegovic, F. Pro, and M. Poncino. Energy-aware design techniques for differential power analysis protection. In *Proceedings of the 40th Conference on Design Automation*, pages 36–41, Anaheim, CA, USA, June 02 - 06 2003.

[6] A. Boukerch, L. Xu, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Comput. Commun.*, pages 11–12, September 2007.

[7] D. Brumley and D. Boneh. Remote timing attacks are practical. In *Proceedings of the 12th Conference on USENIX Security Symposium*, volume 12, Washington, DC, August 04 - 08 2003.

[8] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, May 2003.

[9] Y. chun Hu, A. Perrig, and D. B. Johnson. Wormhole detection in wireless ad hoc networks. Technical report, 2002.

[10] M. Demirbas and Y. Song. An rssi-based scheme for sybil attack detection in wireless sensor networks. In *Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks*, pages 564–570, 2006.

[11] J. R. Doceur. The sybil attack. pages 251 – 260, 2002.

[12] W. Du, J. Deng, Y. Han, and P. Vashney. A pairwise key predistribution scheme for wireless sensor networks. In *ACM CCS*, pages 42–51, October 2003.

[13] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, pages 41–47, November 2002.

[14] P. Gupta and P. Kumar. Capacity of wireless networks. Technical report, University of Illinois, Urbana-Champaign, 1999.

[15] K. ho Lee, H. Jeon, and D. Kim. *New Technologies, Mobility and Security*, chapter Wormhole Detection Method based on Location in Wireless Ad-hoc Networks, pages 361–372. Springer Netherlands, 2007.

[16] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, May 2003.

[17] P. Kocher, R. Lee, G. Mcgraw, and S. Ravi. Security as a new dimension in embedded system design. In *Proceedings of the 41st Design Automation Conference (DAC '04)*, pages 753–760. ACM Press. Moderator-Srivaths Ravi, 2004.

[18] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal Selected Areas in Communications*, pages 1265–1275, September 1997.

[19] P. Malagon, J. C. Vallejo, J. M. Moya, A. Araujo, and O. Nieto-Taladriz. Dynamic environment evaluation for reliable AmI applications based on untrusted sensors. In *The International Conference on Emerging Security Information, Systems, and Technologies*, pages 128–131. SECURWARE 2007, 2007.

[20] R. D. P. Mauro Conti and L. V. Mancini. Ecce: Enhanced cooperative channel establishment for secure pair-wise communication in wireless sensor networks. *Ad Hoc Networks*, 5:49–62, January 2007.

[21] D. Mukhopadhyay and I. Saha. Location verification based defense against sybil attack in sensor networks. In *ICDCN 2006, LNCS 4308*, pages 509–521, 2006.

[22] J. Newsome, E.Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 259–268, 2004.

[23] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady. Security in embedded systems: Design challenges. *Trans. on Embedded Computing Sys.*, 3(3):461–491, 2004.

[24] S. Ravi, A. Raghunathan, N. Potlapally, and M. Sankaradass. System design methodologies for a wireless security processing platform. In *Proceedings of the 39th Conference on Design Automation*, pages 777–782, New Orleans, Louisiana, USA, June 10 - 14 2002.

[25] N. Subramanian, C. Yang, and W. Zhang. Securing distributed data storage and retrieval in sensor networks. *Pervasive and Mobile Computing*, 3:659–676, December 2007.

[26] I. Vajda, L. Buttyán, and L. Dóra. Statistical wormhole detection in sensor networks. In D. W. Refik Molva, Gene Tsudik, editor, *Lecture Notes in Computer Science*, pages Volume 3813/ 2005, pp. 128 – 141. Springer-Verlag GmbH, 2005. Security and Privacy in Ad-hoc and Sensor Networks: Second European Workshop, ESAS 2005, Visegrad, Hungary, July 13-14, 2005.

[27] W. Wang, B. Bhargava, Y. Lu, and X. Wu. Defending against wormhole attacks in mobile ad hoc networks. 2002.

[28] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 2002.

[29] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *Parallel Distributed Computing.*, pages 110–122, February 2003.

[30] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning. Defending sybil attacks in sensor networks. In *Proceedings of the International Workshop on Security in Distributed Computing Systems*, June, 2005.

Towards User-centric Identity Interoperability for Digital Ecosystems

Hristo Koshutanski
Computer Science Department
University of Malaga (Spain)
Email: hristo@lcc.uma.es

Mihaela Ion
CREATE-NET Research Center
Via alla Cascata 56/D, Trento (Italy)
Email: mihaela.ion@create-net.org

Luigi Telesca
CREATE-NET Research Center
Via alla Cascata 56/D, Trento (Italy)
Email: luigi.telesca@create-net.org

***Abstract*— Digital Ecosystem is a new paradigm for dynamic IT business integration. Its main focus is to provide micro- and small enterprises with technological solutions bootstrapping their growth and cooperation. In a Digital Ecosystem, institutions compete in some business aspects and collaborate in others, and thus form stable and unstable coalitions. Such a dynamic environment becomes a bottleneck for identity management solutions. Existing and well-researched solutions for identity federation are either too restricting and not flexible enough to support the dynamic nature of ecosystems or they are too complex and difficult to adopt by small enterprises.**

In this paper we present a model targeting cross-domain identity interoperability between distributed ecosystem entities. The model is based on the recent OASIS SAML v2.0 standard to provide interoperability and convergence between existing identity technologies. The paper presents the basic and extended identity models for single services and service compositions. The aim of this research is to allow small and medium companies to use and enhance their current identity technology with a practical and easy to adopt identity management solution that scales up to the dynamic and distributed nature of digital ecosystems.

Keywords: Identity management, Single-sign on, Digital ecosystems, Identity interoperability, User-centric identity profile.

I. INTRODUCTION

Digital Ecosystem (DE) [14] is an innovative multidisciplinary concept that explains how dynamic business coalitions can be supported through an open IT environment. DE are a set of open standards, joint infrastructure and advanced services supporting the dynamic evolutions of business relations and virtual organizations over time.

DE complements current Service Oriented Architectures (SOA) with a sustainable approach that overcomes the limitations of SOA. SOA provides service composition opportunities only though a high level ”centralized” architecture and broker/integrator coordination. SOA/WS standards can, in fact, facilitate services integration only in the context of a well defined business domain where a big player can dictate certain rules, standards (even proprietary) and/or basic communication conditions.

However, software interoperability and integration are totally different in the context of a whole industry, where a role of business broker is not well specified in advance and can change over time based on market conditions and target markets. In this context, establishing a certain level of control and coordination, even in modest amounts, is very difficult. The justification for this non efficient behavior is motivated by the fact that usually integrators try to lock in other players (usually suppliers) in order to avoid vertical and even horizontal competition controlling the supply chain and the evolution of the target markets.

Digital Ecosystem aim to overcome those limitations. DE provides a Peer to Peer (P2P), interoperable, service infrastructures supporting the dynamic nature of the business ecosystems. DE is therefore an open, decentralized, communication and service infrastructure populated by networked agents (big and small and medium enterprises, service brokers, public bodies, end users), data, knowledge models and software services supporting the interaction of the above mentioned ”species” and the evolution of the open ecosystem. Thanks to this open (joint ownership of the infrastructures) and friendly approach DE provides new infrastructure enablers that can facilitate the fast deployment of services by Small and Medium companies (SMEs) or even individuals.

In such dynamic environment agents are able to evolve dynamically through incessant transactions, alliances, adaptation and composition of service offerings. They negotiate (cooperating and competing) with the final objective to survive to market competition while increasing their wealth (not only increasing the profits) and competitive advantage. Thanks to the DE approach economic actors can perform different roles (services producers and consumers) over time and with their active participation they can open new market opportunities, business models and service deployment methods. Figure 1 shows the high-level stack view of a Digital Business Ecosystem [15].

Current closed federation approaches are too restrictive and not sustainable over time to support unstable alliances and virtual coalitions. Those solutions are also very complex and not affordable by SMEs. In this dynamic context, identity management solutions need to be more open and easy to use and they should be able to connect entities coming from different business domains and using different certification schema.

Ecosystem-oriented architectures. Today users and organizations employ a broad set of digital components, such as software products, business services, knowledge (documents,

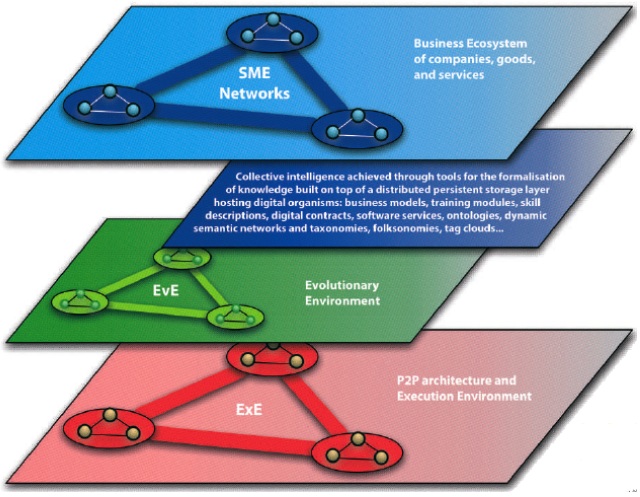


Fig. 1. The stack view of the Digital Business Ecosystem [15]

e-mails, portals, wikis, etc.) and data structure representing business objects. An Ecosystem Oriented Architecture (EOA) [4] can be defined as a meta-level architecture for DE, allowing for the description of digital components and processes that are involved. The idea behind EOA is the extension of the classical SOA in a distributed and semantic rich architecture designed to support the interoperability and the integration of the different processes that characterize a DE. In an EOA all the components interact together, crossing organizations' boundaries and forming a DE that connects different systems, and exchange information using common data representations, like XML and other standard formats. All the EOA services are deployed on a distributed, peer-to-peer platform and described by business and functional models, using Unified Modeling Language (UML), adding in this way semantic to the service description.

This decentralized architecture defines a topology and a replication schema that depend on a set of collaborative peer nodes. A peer-to-peer network supports this topology, and the data replication across the network is guaranteed by a *Distributed Knowledge Base* (DKB) that stores and retrieves contents in a smart way. The final picture is a peer-to-peer and service oriented architecture with high integration capabilities offered by the adoption of open standards where the gap between business abstraction and software implementation is bridged by the adoption of model driven methodologies.

Identity technologies. Institutions use different types of authentication and and identity certificate technologies such as X.509 [26], SPKI [17], Kerberos [11], SAML [16] etc, which are not always compatible and interoperable with each other. Users often need to access applications, services or a composition of services located at different administrative domains.

WS-Policy [24], WS-Trust [25] and WS-Federation [23] cover a wide range of requirements and at the same time are difficult to suit immediately for small and medium size enterprises (SMEs). Existing standards are heavy and difficult to understand, and require a high-cost and longer term deployment, and therefore suitable for large enterprises.

What SMEs in DEs need is a simple and easy do adopt model that allows them to enhance their current identity technology with an extension to identity interoperability management [12].

Paper contribution. Our approach aims at automating the process of identification between ecosystem partners. We emphasize on practical solutions which are clear and easy to implement. The model is based on the new SAML (v2.0) standard [16] for providing proper identification. SAML faces interoperability on the message level and helps to automate and converge when the technologies are not compatible. We face distributed identity storage by the use of user profiles. A user profile is an abstract view of a client's identity information that is stored in a decentralized manner. Decentralization is faced by use of peer-to-peer replication of user profiles on trusted nodes, part of DKB.

The paper is organized as following. Section II defines the core model functionalities scaling to DE's nature. Section III presents the model architecture with its message flow, and the model extension to service compositions. Section IV presents details of a possible user profile structure. Section V discusses the concept of token transformation for interoperability with its inherent SAML-based functionality. Section VI overviews current identity management standards, and Section VII concludes the paper.

II. Identity Model Functionality

Let start by summarizing the main functionality an identity management model for DEs should cover.

1) Dynamic trust relationship establishment and management between identity providers across administrative domains. Identity providers should flexibly define (new) trust relations with other identity providers and the relations should be easy to discover (by end-users) to allow for dynamic, on the fly, trust discovery. The dynamic definition of trust relationship will allow identity providers to maintain and update when their trust relations change, which is often the case in DEs.
2) Enable single sign-on mechanism on top of the established trust relations for decoupling a service provision logic from an authentication (identification) process.
3) Allow user-centric identity management of available credentials for easy discovery and identification to third party service providers/administrative domains.

A. Single sign-on mechanism for identification abstraction

Single sign-on (SSO) mechanism has been developed to provide separation and abstraction between a service provision logic, and an identification process to service users. The abstraction aims at encapsulating the management of an identification process by a third-party called an identity provider (IdP). By adopting an SSO mechanism a service provider (SP)

offloads the burden for a proper user identification to a trusted IdP.

When more than one SPs share a common IdP they form a simple form of identity federation where a user with a single sign-on can access all services under the federated SPs.

Although a SP defines a trusted IdP, still a SP provides access to resources based on a simple form of user authentication. The SP's authentication process is a verification process of whether a user has been identified by a trusted IdP. Generally, an SSO mechanism can be initiated by both a SP or a user accessing a service. The user-initiated SSO allows users to select a user-trusted IdP that a SP should contact for a user identification. Even though the user-initiated SSO has several application scenarios for Web-based authentication (e.g., OpenID SSO mechanism[1]), it is not suitable for our purposes.

Our targeted SSO is the SP-initiated SSO that allows a SP to define and maintain its own (federated on not) IdPs. Below we describe the SP-initiated SSO case.

1) A user is accessing a service under a SP without any login information.
2) The user is not recognized and gets redirected to a SP-trusted IdP.
3) The user is required to sign on by providing required credentials to the IdP (e.g., user name and password or an identity certificate).
4) If successful authentication, the IdP redirects the user back to the SP including information about the user authentication, often in a form of a security token.
5) The SP, in its turn, verifies if the user authentication has been done by the IdP (a simple authentication process if an IdP has digitally signed user authentication information), and gives access to the requested service. A security context (session) is created based on the user authentication.

Let us start by defining the key components of the model.

1) *User*: any entity that can be identified in the network (peer or web browser user, institution or person)
2) *Service Provider (SP)*: any identifiable entity that has one or more services or resources available to other entities.
3) *Identity Provider (IdP)*: any entity that is able to provide digitally signed credentials to other entities.
4) *Digital Ecosystem (DE)*: distributed digital environment where both partners and competitors are present and where stable and unstable coalitions are created; coalition of digitally represented partners with few or no a priori established trust relations. Thus the notion of ecosystem comprises cooperative and competitive relations.

We target an identity management model for decentralized peer-to-peer ecosystem domains. All entities are considered equal and there is no hierarchy of ecosystems. Any peer can be an IdP or a SP, or both. Each user can issue a certificate to other users. Each user has a list of trusted IdPs. Each IdP has a list of acceptable security tokens. An IdP issues certificates to users based on:

- security tokens issued by the provider itself, or
- security tokens issued from IdPs with whom it has trust relationships, or
- user registration information.

B. Multiple user identities and technology standards

In a network of interconnected digital ecosystems, users and companies use different kinds of certificates obtained from outside the system. Companies have own X.509 certificates issued by Certification Authorities outside the system and which they are obliged by law to use when doing online transactions. SMEs often have their own proprietary solutions for identification of their employees such as user name and password, ad hoc secure tokens or adoption of OpenID for Web-based access.

After joining a DE, users (partners) obtain a variety of certificate tokens issued by IdPs for particular business needs. However, partners that already have ad hoc identity tokens or user name/passwords authentication should be able to use them for the sake of providing identity information to IdPs that are to certify partners' identity. The reason for that is to unify identity management between partners with already existing identity token standards.

Each IdP has the responsibility to provide proper pseudonymity to end users. An IdP either issues a user pseudonym on its own or allows users to define it and then certifies the pseudonym in a security token to a SP. A SP explicitly asks an IdP to reveal user identity in case of user misbehavior.

C. User-centric identity profile

Having multiple identity certificates issued by different IdPs, it becomes difficult for a user to manage and locate all of them when needed to access a service, especially in the case of distributed services.

Users connect to a DE either via a portal (a Web browser) or via a rich client system installed on their computers. In either of the cases a user needs a way to manage its credentials, user name/passwords and public/private key pairs. For that purpose we adopted the use of a *user profile*. A user profile contains all available information about user's identity obtained from the user's interactions within DEs. Its main purpose is to provide an abstract view of what identity credentials are available, where they are available (e.g. local or remote storage) and how to obtain them (e.g. via authentication to an IdP by user name/password or via an LDAP[2] storage etc).

An important issue is how to allocate, store, and retrieve the user profile. The profile contains sensitive information that is necessary when communicating with entities in a DE. So, the profile must be protected from unauthorized access (no one except the owner of the file) and at the same time must be

[1] http://www.openid.net

[2] http://www.openldap.org

available on demand (avoid denial of service/availability). To address these issues we adopted to keep the profile encrypted and replicated on trusted peers. The encrypted profile is only meaningful to its owner and reliably obtained via a trusted peer-to-peer network as part of the DKB in DEs.

Another issue worth mentioning is the availability of a profile to be shared (used) by multiple entities. This may often be the case for SMEs where selected employees are allowed to use the profile and therefore represent the company in on-line business negotiations. A possible approach is to define an access policy for each profile that encodes who can use the profile and under what conditions. The access policy is optional and if not explicitly specified it should have the default value of only read and write permissions for the profile's owner. A simple and yet effective solution for the policy model is the use of Access Control Lists[3].

We adopted the concept of peer-to-peer trusted network, as provided by the DKB of DEs infrastructure, to replicate and provide service availability when locating and loading user profiles. The problem of how to establish a proper methodology for data replication is beyond the paper scope, and some works can be found in [13], [22].

A user is required to remember a user name and a password in order to login into a DE. The user name and password are obtained on initial user registration to a DE. Once registered, whenever the user logs in another (or a same) DE with its login information, the DE's infrastructure takes the responsibility to allocate and retrieve the encrypted user profile.

When a user starts a new session, its profile is to be downloaded on a secure memory (e.g. browser s-box) of its Web browser or a local client and then decrypted. Once decrypted the profile is ready to be used and processed by the Web browser client or the local client. On end of a session, the user profile is encrypted again and updated on the associated trusted node (peer) and then replicated on other trusted peers.

In the case of a local client installed on user's own machine, the profile could be locally copied and stored so that it could be loaded from the client's machine next time. However, in this case the profile must also be stored and replicated on other trusted peers in order to provide availability and actualization if shared among multiple users.

The user profile is encrypted with a long master password, usually a key phrase, known only to a user. The master password should be different from the user password needed for user authentication to a DE. Thus, a user has to remember one login information and one master password in order to facilitate a secure profile storage.

D. Identity profile evolution over time

A user profile contains information about available identity certificates, public/private key pairs and user authentication information needed to access and obtain security tokens.

User identity information obtained outside DEs should be updated (imported) in the user profile so that it can be re-used when the user does business interactions with partners in DEs. When a user first time registers to a DE and creates its initial profile, it is requested to import the already available identity information. However, a user can start from no identity information and collect it on a step-by-step basis when interacting with SPs and their trusted IdPs.

An important aspect here is the possibility of evolving user's identity token information dynamically, during normal user interactions with ecosystem partners. After each interaction with an IdP, the user's client (web or local) automatically records the information on the new identity token for subsequent use. The information stored should detail the new token, issuer, authentication process used to obtain the token (e.g., by another token authentication or by user name/password), token type, validity and location of token retrieval, refer also to Section IV. This would allow users to dynamically discover new trust relationships between IdPs and obtain the respective identity tokens for proper authentication, as discussed later in Section V-C.

E. Token transformation for interoperability: SAML approach

To approach proper identity management first we need to define a way to cope with the incompatibility of the variety of standards and solutions. Here we borrow the concept of credential transformation from one type to another as already introduced in the WS-Trust standard. To address the problem we have to convert identity information from one certificate technology to another one compatible with the current domain of business.

We have to provide a way for a client identified within one standard to be able to use its identity information when communicating with a SP using another identity representation standard. The issue we take into account is that SMEs may adopt their own (ad hoc) certificate tokens or mechanisms to manage identities of their employees.

To cope with this wide range of identity mechanisms we make the following assumptions.

- Each SP adopts the identity standard best suiting its needs but its trusted IdP should support as a default authentication the SAML standard (especially v2.0). It means that a SME could preserve its existing identity management infrastructure but should enhance its trusted IdP to be SAML-aware, i.e. the IdP should issue SAML authentication assertions derived (transformed) from any of the standards the IdP already supports.
- In order to provide a correct semantic identification and processing between different identity technologies we also impose a SP to be SAML-aware (as a default setting) for its interactions with a trusted IdP. In this way, each SP adopts SAML identity assertions as means of proof of identification between the SP and its trusted IdPs. For example, a transformation of SPKI to SAML and then of SAML to X.509 may be semantically incorrect due to the different design goals behind the X.509 and SPKI standards.

[3]http://en.wikipedia.org/wiki/Access_control_list

With the new SAML release, the standard allows to express identity assertions within a context of many types of authentication, such as X.509, SPKI, Kerberos tickets, user name/password, etc. Thus, SAML becomes a suitable message format standard for unifying identification information of different identity standards. SAML authentication assertions are used when accessing or negotiating with different ecosystem domains.

For example an IdP that supports X.509 and user name & password authentication to be functional/compatible in our framework it has to also support the following authentication to SAML-based conversion:

- X.509 token-based authentication to SAML identity assertion
- User name & password authentication to SAML identity assertion.
- SAML-based authentication to SAML identity assertion.

Example 1: Let us suppose that a SP_1 only accepts X.509-based authentication to identify entities and that SP_1 trusts IdP_1 to validate and authenticate users based on X.509 tokens. Now, let IdP_2 identifies users based on SPKI tokens, and let a user has a SPKI certificate issued by the IdP_2. If IdP_1 and IdP_2 have bilateral contractual relationship of sharing users' identities for the sake of common service usage, following our model assumptions, the following conversions are to be provided:

IdP_1: X.509 token-based authentication to SAML identity assertion.
IdP_1: SAML-based authentication to SAML identity assertion.
SP_1: SAML-aware for a proof of authentication from IdP_1.
IdP_2: SPKI token-based authentication to SAML identity assertion.
IdP_2: SAML-based authentication to SAML identity assertion.

With the above assumptions the user is able to automatically identify itself to SP_1. To do so, the user has to contact its IdP_2 and request for a SPKI-based authentication to SAML identity assertion transformation. Based on the model assumption, IdP_2 provides a remote authentication (e.g., SPKI-based with challenge/response) in order to properly authenticate the user. Based on the authentication information and user identity in the SPKI certificate, the IdP_2 digitally signs a SAML authentication statement with the result of the authentication, and returns it back to the user. The user forwards the newly obtained token to IdP_1.

Since IdP_1 has a contractual trust relationship with IdP_2, IdP_1 accepts the SAML assertion, by verifying its signature, and issues a new SAML assertions to SP_1 for proof of entity identification. SP_1 is SAML-aware and trusts IdP_1 for identifying entities and provides access to the desired service.

III. Identity Management Model Architecture

Figure 2 shows the basic model architecture and workflow of messages between the main actors. The message flow of the model is the following:

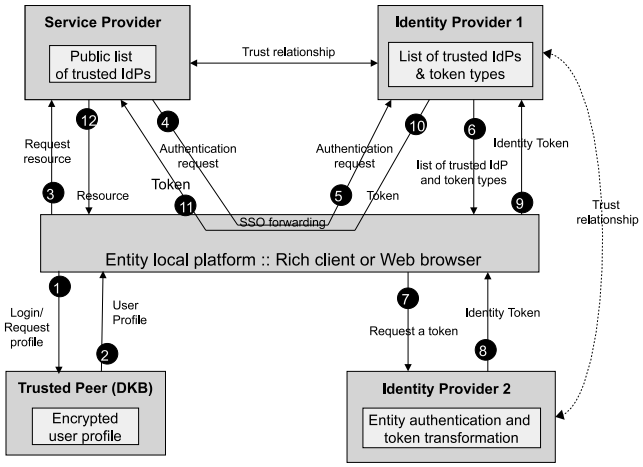


Fig. 2. Model architecture and communication scheme

1: A user requests its profile from a trusted peer storing it by authenticating himself with the user name/password from the registration process. Information of ecosystem trusted peers is obtained (possibly publicly available) when users join the ecosystem.
2: On successful authentication, the trusted peer retrieves and sends the encrypted user profile.
3: The user decrypts the profile (with the master password) and starts using ecosystem services. It makes a request to a SP to access a service.
4–5: The SP redirects the user to a trusted IdP_1 (SSO use case).
6: The user has no credentials issued by the IdP_1. The IdP_1 sends a list of its trusted IdPs and the accepted token types to the user.
7: The user queries the profile if it has available tokens. The profile is processed to match if there are tokens (information) issued by any of the IdPs from step 6. If no credential is matched then the user (possibly) has to register to IdP_1 to obtain an identity token. If an identity token is found then the user extracts it either from the profile, or requests it from the remote IdP that issued it. If a match of IdP and token type then the user just presents the certificate as it is. In case of more than one possible matches the user is prompt to choose which token to use. We note that the user can perform steps 8 and 9 even if it has the right credential match of IdP and token type but does not have the token locally in the profile. In such a case, the user obtains it via a remote authentication (e.g., LDAP server storage).
8: If a credential match, e.g., of IdP_2 but with different token type then the user requests IdP_2 for authentication and transformation to a SAML identity assertion. On successful authentication, IdP_2 issues a SAML assertions and returns it to the user.
9: The user forwards the certificate/SAML assertion to IdP_1.

10: IdP_1 verifies and validates the certificate and issues a SAML assertion to be forwarded to the SP.
11: The user is redirected to the SP which accepts tokens from IdP_1.
12: The SP verifies the new certificate and provides the requested resource to the user.

We note that in step 10, IdP_1 certifies the authentication outcome in step 9 with an identity token acceptable by the SP. The only case in which IdP_1 does not issue a new token is when the user (already been in contact with the SP) presents a same identity token issued by the IdP_1 last time. In this case, after certificate verification and validation, IdP_1 forwards the token to the SP.

The SAML standard is to be used when user authentication format with an IdP is different than the one agreed between the SP and the IdP. For example, in case of X.509 user authentication to IdP_1 and a same X.509 format agreed between the SP and IdP_1 then, IdP_1 may not issue a SAML authentication statement to SP but use the X.509 format.

Step 11 is a point where the user profile records and stores the new identity token and associated information for a subsequent use.

A. Model extension to service composition

Digital Ecosystems allow companies to cooperate with each other, form coalitions, and thus use service compositions suitable for their business models. An important requirement for an identity management model is to support composition of services. We extend the basic model presented above to cope with the case in which one service relies on services from other providers. We assume that the service composition model occurs between SPs having contractual trust relationships.

In a service composition scenario, the service provider aggregating services from other service providers needs to run the services on the name of the user and, as so, he has to authenticate the user to the other providers. To solve this problem we adopted the use of *Proxy Certificate* that the client issues to the provider of the composite service.

A Proxy Certificate [18] is derived from and signed by a normal X.509 public key end-entity certificate or by another Proxy Certificate (PC). The identity of the new PC is derived from the identity that signed it. A PC has its own public and private key pair. A PC is identified as such by its extensions. Any X.509 certificate has extension fields to encode different certificate characteristics. A PC has a policy that specifies what conditions must be respected when an entity is using it. Another important issue is that a PC can only sign another PC.

SAML standard v2.0 defines a rich set of subject classification, as part of the SAML identity assertion, that allow entities to be bound to a public-key information. In this case, the real use of proxy certificate in our model is in the SAML context definition. Thus, the message encoding of identity information in a proxy certificate becomes as a SAML assertion and is compatible (message-level interoperable) with the SPs of composite services.

There are two important requirements specified in the policy of a proxy certificate that reflect our identity model.

- *Service scope.* The first requirement is the scope of a PC. We identity the scope of a PC to be the scope of the service being requested by a client. Scope of service means any aggregated service that is directly used for the sake of proper execution of the main service. In other words, any service that is not directly aggregated within the main service (e.g. aggregation of aggregation, or third-party services not part of current aggregation) should fall beyond the PC scope, i.e., not considered as a valid identity certificate on behalf of a client.
- *Level of service aggregation.* To solve the issue of complex aggregation of services that aggregate other services, we propose as a second requirement the level of service aggregation. The purpose of the level of aggregation is to restrict the use of a PC in a chain of service aggregations. Often a client may wish to restrict not only the scope but also the re-generation of next level PCs. For example, to restrict the use of a service of selling books, a user may use level of aggregation 1 indicating that the at most one level of aggregation is allowed, expecting only a product shipping service to be used and not further delegation of PC usage. The level of aggregation should be interpreted as not to derive more PCs longer in chain than the specified level.

Another requirement is a validity period of a PC. Usually, this depends on the particularity of the main service being executed (i.e., the validity of the service transaction). The client obtains such information from the SP hosting the main service. This parameter plays an important part of PC usage. A client may restrict the use of a PC according to his expectations or familiarity with a given SP. If a distrusted SP a client may wish to generate a PC with short validity period to reduce potential misuse of it.

When a SP contacts another SP to execute an aggregated service, the second SP specifies that it needs a PC to execute other services within its aggregated service. To do so, the first SP issues and signs a new PC to the second SP with the following restrictions:

- The derived PC has as service scope the aggregated service to be executed,
- The derived PC has a level of aggregation the level of the predecessor PC decreased with 1 (if not zero),
- The derived PC has a validity period, the remaining validity period of the predecessor PC that signs it (if not expired).

In this way, a next level SP can use the newly derived PC only for the sake of execution of its service. To validate a PC an IdP needs the set of all PCs derived from the client's generated one, and validates if they follow a correct PC derivation as described above.

Figure 3 shows the extended identity model for service compositions. The steps behind the model are the following:

1: The user downloads the profile from a trusted peer.

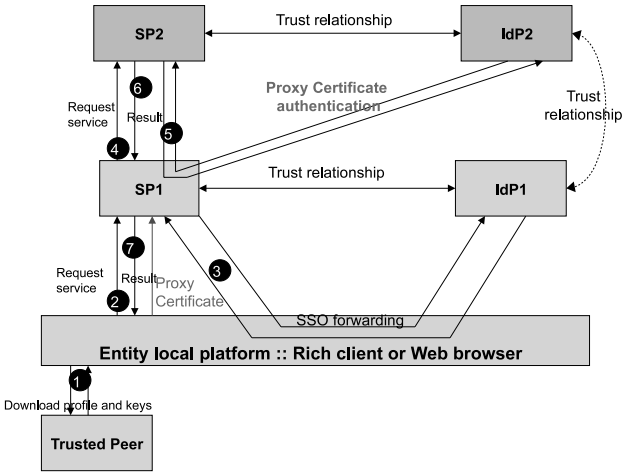


Fig. 3. Service composition using proxy certificates

2: The user requests to access a composite service of SP_1.
3: The user is redirected to IdP_1 to login (SSO use case). On successful authentication following the message flow of the main model, IdP_1 issues a new (SAML) identity token forwarding it back to SP_1. SP_1 indicates to the user that the requested service is an aggregation of services together with a list of the services to be used. The list of aggregated services is an optional (but recommended) element and it serves to precisely define in a PC the service scope. The default value of the service scope is the current service the user requests with a level of aggregation 1. The user issues a PC to SP_1.
4: SP_1 requests a service to SP_2.
5: SP_2 redirects SP_1 to IdP_2 for user authentication. SP_1 authenticates to IdP_2 on behalf of the user using the proxy certificate obtained in step 3. We note that a new PC has its own private/public key pair used for an authentication process. For successful authentication, SP_1 sends to IdP_2: *(i)* the PC issued from the user, *(ii)* the identity token received from IdP_1 for user authentication, and *(iii)* the user original certificate that signed the PC. The last certificate is essentially the one the user has authenticated with to IdP_1.
We made the assumption that service aggregation occurs between contractual trust relationships, in this case, IdP_2 has a trust relationship with IdP_1 and can validate that the original certificate which signed the PC has been used for authentication by IdP_1, by analyzing the identity token issued from IdP_1, and that the PC remains valid according to its specified policy. If successful verification, IdP_2 issues an identity token to SP_2 (binding the original user identity) authorizing SP_1 as running on behalf of an authenticated user.
6; SP_2 runs the service and provides the result to SP_1.
7: SP_1 completes the service execution and provides the result to the user.

In case of more than one SPs on a next level aggregation, e.g., $SP_2, SP_3, ..., SP_n$, the aggregator SP issues a new PC for any of the SPs on the next level, by repeating the extended model n-times. The extended model scheme can be recursively applied in case SP_2 needs to contact SP_3 as next level aggregated service provider. In such a case, SP_2 takes the role of SP_1 in the extended model.

B. Model integration in DE

Ecosystem oriented architectures [5] provide specific mechanisms for peer-to-peer decentralized communications. There is an abstraction communication layer that, close to Grid communications, defines seamless and platform independent service provisioning and execution. In this way ecosystems' services interact with each other transparently of the communication layer and regardless whether service provisioning and execution takes place on a remote or local platform.

Each SP, providing services via the DE's infrastructure, defines a trusted IdP (or a list of them), as a dedicated DE's service so that DE users will be forwarded to it. On the other side, each DE user will benefit of the DKB part of the DE's infrastructure for a distributed storage and retrieval of its profile. The DKB accessibility service requires a user to be a registered DE user, which will bootstrap the identity management model with a token availability from the initial registration. This is especially useful for those IdPs that accept DE's registration tokens for possible user authentication.

An SSO is the main interaction mechanism for a user authentication. An IdP has two main services relevant to the proposed model: an authentication service and a transformation service. Since the services an IdP provides are DE's dedicated services, the SSO mechanism (between an SP and an IdP) is to be based on top of the DEs communication layer. Thus, a user will use the DE's standard mechanism for service accessibility in both when requesting a secure token transformation and when authenticating to an IdP.

A comprehensive identity management solution for DEs is tightly bound to the definition of suitable trust and reputation mechanisms that benefit (are based upon) the identity model. The work in [9] defines a peer-to-peer reputation framework for quantifying trust on different levels of DEs stepping on the identity model in [12]. The work in [10] complements [9] by presenting an agency-based reputation model as a more reliable trust quantification schema. The agency reputation model defines an interoperation schema between agencies to provide a scalable reputation solution to DEs. Our aim with the above approaches is to define a targeted trust and identity management framework for DEs that scales to the needs of SMEs.

IV. User-centric Identity Profile

In this section we will describe the structure and syntax of a user profile.

User profile structure. The user profile is built using RDF (Resource Description Framework) meta-model and XML syntax [20]. RDF provides a language for representing information and information modeling. RDF works on the basis

of making statements about resources. These statements about resources are given in the format of subject-predicate-object [19]. The subject refers to the resource, predicate refers to a property or aspect of the subject, and object assigns a value to this predicate.

RDF identifies entities using Web identifiers (called Uniform Resource Identifiers, or URIs), and describes resources in terms of simple properties and property values. This allows RDF to describe statements about resources as a graph of nodes and arcs representing the resources, and their properties and values.

The user profile has a generic structure of:

- A standard v-Card format, and
- A listing of relevant identity token information available to a user.

The user profile lists all available security tokens together with relevant token information. The user profile provides a unified view of the user's identity information. Users get certificates from interactions with different DE domains. The user profile will be referenced across the DKB as an I-name XRI reference [2]. The following example of an RDF graph depicts the structure of the user profile:

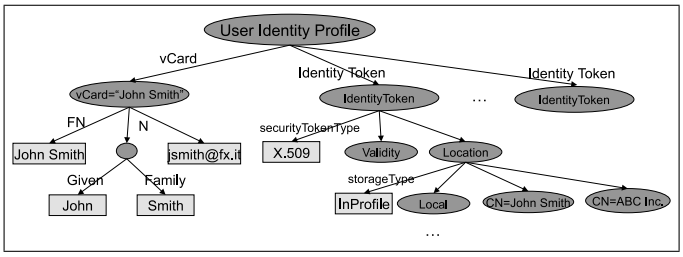


Fig. 4. User profile example

The JENA framework[4] provides a programmatic environment for reading, writing, querying and updating RDF documents in several formats such as RDF/XML and Turtle. We will overview the functional description of the core classes used in the profile structure.

- `UserData` A class encapsulating basic user information provided during a registration process. It corresponds to the vCard information in the profile.
- `Profile` A class encapsulating the identity profile that contains information about all credentials a user has.
- `IdentityToken` A class encapsulating/corresponding to one credential entry in a user profile.
  - `TokenType` – X.509, SPKI, SAML assertion, user name & password, etc.
  - `Subject` – subject name of the user in a given token, i.e. how the user is known to a given IdP.
  - `Issuer` – issuer distinguished name as defined in an identity token.
- `Location` A class encapsulating a location of a certificate. Token availability in a profile and how to access it. If the token type is user name & password, then the token will be contained in the profile. If a different token type, the token will be either stored in an PKCS#12 attached to the RDF profile or will be stored on an external LDAP (Lightweight Directory Access Protocol) server.
- `Validity` A class containing a validity period of an identity token, in a format `notBefore` and `notAfter` dates.
- `AccessInfo` A class encapsulating the information on how to access/retrieve a certificate. Information about the location of the server (URI), the location of the certificate (distinguished name: DN) and possibly user name and password information for accessing the token will be available in the profile.

Based on the core classes the following methods/interfaces are provided for a user profile management.

- `createProfile(UserData)` Creates a user profile based on the information in class UserData. It creates a registration data such as name, address, email, etc. It creates also the vCard as described in the RDF schema.
- `addIdentityToken(IdentityToken)` Adds an identity token information to a current user profile.
- `deleteIdentityToken(IdentityToken)` Deletes a credential information from a current profile. This usually happens because a certificate has expired or an Identity Provider leaves a network or is no longer trusted.
- `matchIdentityTokens(List_of_Trusted_IdPs)` Returns a list of matched IdentityToken elements. It queries a current user profile for all credentials matching an IdP and a TokenType from the input list. Later in the section we will describe what data structure an IdP returns to a client for a list of trusted IdPs used for the query process.

Identity credential token schema. While the vCard structure is obviously a syntax supported by a standard schema [21], the identity token syntax needs to be defined within the RDF structure. The RDF graph for an identity credential token is depicted in Figure 5.

We will use Turtle [3] for expressing the structure of the RDF schema. Turtle allows RDF graphs to be written in a compact and natural text format. The listing below shows the identity token schema:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://www.one-node.org/2008/04/profile> .

  :IdentityToken a rfds:Class .
  :Validity a rfds:Class .
  :Location a rfds:Class .
  :AccessInfo a rfds:Class .

  :securityTokenType a rdf:Property ;
        rdfs:domain :IdentityToken ;
        rdfs:range [
          a rdf:Alt;
          rdf:_1 rdfs:datatype("X509", xsd:string) ;
          rdf:_2 rdfs:datatype("SAML", xsd:string) ;
          rdf:_3 rdfs:datatype("SPKI", xsd:string) ;
          rdf:_4 rdfs:datatype("UsrnPswd", xsd:string) .  ] .
```

[4]http://jena.sourceforge.net

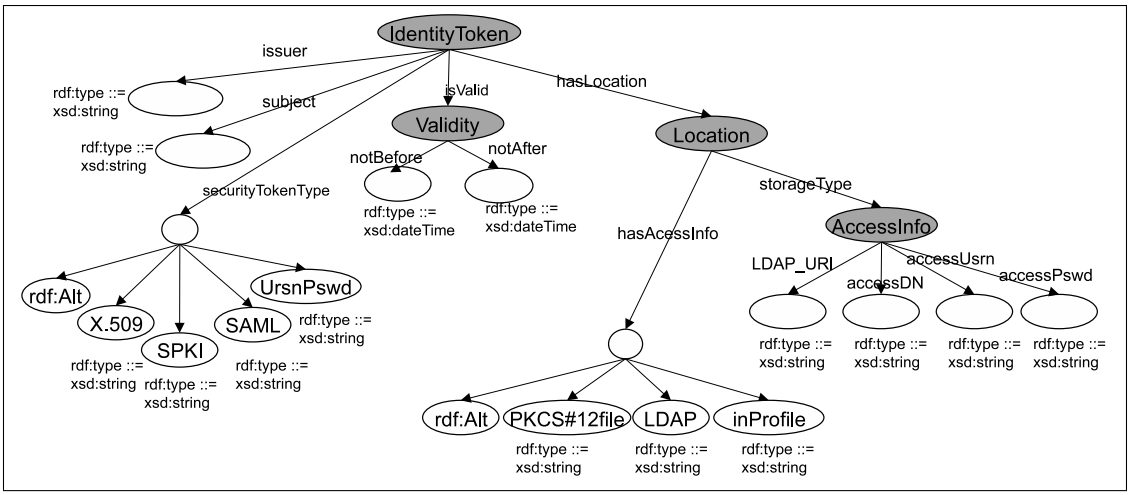


Fig. 5. User profile identity token schema: an RDF graph

```
:isValid a rdf:Property ;
      rdfs:domain :IdentityToken ;
      rdfs:range :Validity .
:notBefore a rdf:Property ;
      rdfs:domain :Validity ;
      rdfs:range xsd:dateTime .
:notAfter a rdf:Property ;
      rdfs:domain :Validity ;
      rdfs:range xsd:dateTime .
:issuer a rdf:Property ;
      rdfs:domain :IdentityToken ;
      rdfs:range xds:string .
:subject a rdf:Property ;
      rdfs:domain :IdentityToken ;
      rdfs:range xds:string .
:hasLocation a rdf:Property ;
      rdfs:domain :IdentityToken ;
      rdfs:range :Location .
:storageType a rdf:Property ;
      rdfs:domain :Location;
      rdfs:range [
        a rdf:Alt;
        rdf:_1 rdfs:datatype("PKCS#12file", xsd:string) ;
        rdf:_2 rdfs:datatype("LDAP", xsd:string) ;
        rdf:_3 rdfs:datatype("inProfile", xsd:string) .   ]
:accessInfo a rdf:Property ;
      rdfs:domain :Location ;
      rdfs:range :AccessInfo .
:LDAP_URI a rdf:Property ;
      rdfs:domain :Location ;
      rdfs:range :AccessInfo .
:ND a rdf:Property ;
      rdfs:domain :AccessInfo ;
      rdfs:range xds:string .
:accessUsrn a rdf:Property ;
      rdfs:domain :AccessInfo ;
      rdfs:range xds:string .
:accessPswd a rdf:Property ;
      rdfs:domain :AccessInfo ;
      rdfs:range xds:string .
```

V. Identity Token Transformation for Interoperability

The main SAML objective is the ability of expressing assertions about a subject in a portable fashion so that other applications across domain boundaries can trust it.

Authentication statements assert to the service provider that the principal did indeed authenticate with the identity provider at a particular time using a particular method of authentication. Other information about the authenticated principal, called the authentication context, can be inserted in an authentication statement.

SAML authentication statement. A SAML authentication statement defines the following triple: ⟨Issuer, Subject, Validity_Period⟩. Interactions between a user and an IdP for a SAML identity assertion transformation occur within a SAML context, i.e. using the SAML authentication request/response protocol.

The authentication process will be based either on an identity token issued by the IdP or a user name and password authentication. For example, if a user has a SPKI token issued by an IdP and the user needs to have a corresponding SAML identity assertion, the user will initiate a SAML authentication request to the IdP. The authentication process will be based on the SPKI token the user has from the IdP (via challenge/response for authenticity). On successful authentication, the IdP will issue a SAML authentication statement with a userID taken from the SPKI token. We note that an optional input to the transformation interface can be provided allowing a user to specify the need of a pseudonym to be used in the new SAML authentication token.

SAML authentication context. A relying party (a SP's trusted IdP) may require information additional to the assertion itself in order to assess its level of confidence in that assertion. SAML does not prescribe a single technology for authentication and it may vary from an IdP's to IdP's policy. For that case a SAML authentication context is provided to specify additional information, to the authentication process generating a current SAML token, such as what authentication mechanism or method (e.g., password or certificate-based SSL) was used.

Thus, in our example, the IdP issuing the SAML authentication token will additionally specify an authentication context as SPKI-based SSL authentication. Based on that information, the relying IdP can infer what authentication took place and generate the SSO token response (to the SP) with longer/shorter session validity period, or even refuse to accept the SAML token.

```
          <List_of_TIdP> ::=  <IdP_def> | <IdP_def> <List_of_TIdP> .
               <IdP_def> ::= <IdP_id> <IdP_accepted_tokens> .
                <IdP_id> ::= [<Public_key_certificate>] <Distinguished_name> [<List_of_TIdP_URL>] .
    <Distinguished_name> ::= <IdP_name_type> <IdP_name_value> .
         <IdP_name_type> ::= "X500" | "I-Name" | "String" .
        <IdP_name_value> ::= <string_value> .
      <List_of_TIdP_URL> ::= <string_value> .
   <IdP_accepted_tokens> ::= <Token_type> | <Token_type> <IdP_accepted_tokens> .
            <Token_type> ::= "X509" | "SPKI" | "SAML" | "UsrnPswrd" .
<Public_key_certificate> ::= <Token_type> <Token_encoding> <Token_value> .
        <Token_encoding> ::= "Base64" | "Binary" .
```

Fig. 6. List of trusted IdPs structure: BNF notation

Some of the possible context authentication schemes relevant for our scope are: SPKI, X.509, Kerberos, PGP, SSL certificate, password, previous session.

A. Token transformation services

We have two main token transformation functionalities. They represent a remote invocation from a user to a trusted IdP server. Essentially, the two functionalities provide a user authentication process via either an identity token (e.g., SSL-based, challenge/response-based) or via a user name&password login.

Token-based authentication to SAML transformation. An interface that transforms from available token formats to a SAML identity token. A user is authenticated based on its available certificate token. On successful authentication the interface transforms the user authentication information to a digitally signed SAML authentication assertion. The interface (optionally) should allow a user to specify an alternative user identity (user-chosen pseudonym) to be bound to the new SAML identity token. This would allow a user to have privacy (to some extend anonymity) in a given domain. If a pseudonym is used in a SAML token the user should not re-authenticate with that token and request for a new pseudonym, i.e. derivation of a pseudonym from a pseudonym should not be allowed.

User name-based authentication to SAML transformation. An interface that transforms a user name to a SAML assertion. If a user name&password match those of IdP's internal database then a SAML assertion is generated with the user name as a user identity in the SAML token. An optional pseudonymity input should allow a user-chosen identity name to be used instead of his original user name in the new SAML assertion. Note that this should not change the original user name of the user but only bound the new user name in the SAML token.

B. Defining a List of Trusted IdPs

Figure 6 shows the core structure used for representing a list of trusted IdPs. A list of trusted IdPs is a set of tuples each identifying an IdP authority. An IdP authority is identified by (optionally) its public-key certificate and by its distinguished name. For each IdP authority identifier we assign a list of accepted security token types from that authority. Note that Figure 6 describes the data structure and not the representation of a list of trusted IdPs.

A suitable representation of the shown structure is in an XML-based format. We assume that there is a commonly shared dictionary between entities for unambiguous processing of the above (labeled) information. If using Web Services technology, a suitable ground for setting up a list of trusted IdPs is the use of WS-Policy framework[5]. WS-Policy provides a set of basic constructs for defining requirements (basic assertions) about service accessibility.

Example 2: (List of Trusted IdPs):

```
<List_of_TIdP>
  <IdP_def>
     <IdP_id>
        <Public_key_certificate>
           <Token_type> X509 </Token_type>
           <Token_encoding> Base64 </Token_encoding>
           <Token_value>
           -----BEGIN CERTIFICATE-----
           MIIB+jCCAWOgAwIBAgICAfQwDQYJKoZIhvcNAQ...
           EENhbGlmb3JuaWEgU3RhdGUxHDAaBgNVBAMT...
           MTQ0NjQxWhcNMDkwOTAxMTQ0NjQxWjA/MQs...
           cmtlcnMgT3JnLjESMBAGA1UEAxMJSm9obiBDb3V...
           -----END CERTIFICATE-----
           </Token_value>
        </Public_key_certificate>
        <Distinguished_name>
            <IdP_name_type>X500<IdP_name_type>
            <IdP_name_value>
              CN=ABC CA Class-1,O=ABC Inc.,C=US
            </IdP_name_value>
        </Distinguished_name>
     </IdP_id>
     <IdP_accepted_tokens>
         <Token_type> X509 </Token_type>
         <Token_type> SAML </Token_type>
     </IdP_accepted_tokens>
  </IdP_Def>
</List_of_TIdP>
```

The example shows an XML representation of a list of trusted IdPs with only one certification authority. The IdP is identified with an X.500 distinguished name, and the accepted security tokens are X.509 and SAML.

C. User profile evolution: Dynamic token discovery

The main feature of a user-centric identity profile is the possibility of dynamic evolution over time. On each interaction with a SP the user profile will update user identity token

[5]http://www.w3.org/Submission/WS-Policy

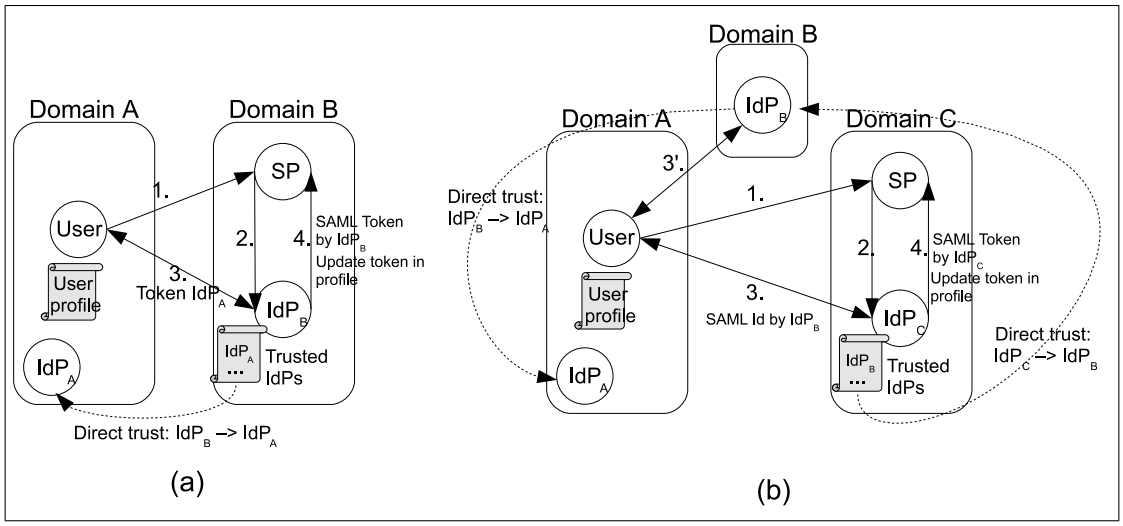


Fig. 7. User profile evolution scenario

information with a new token information obtained from the interactions with the IdP, trusted by the SP. Figure 7 shows the main envisaged scenario.

Figure 7(a) illustrates a basic scenario of a direct trust relationship between a SP's IdP_B and an IdP_A that has issued a token to a user. We represent the two IdPs as belonging to different administrative domains A and B, respectively. The SSO between the user and IdP_B of domain B will authenticate the user by using the existing token information, as already shown in the model. Let assume that the existing token information is of SPKI format and IdP_B accepts only SAML tokens (the default format in the model). After an authentication and a transformation process with IdP_A via the existing token, the user obtains a SAML authentication token that it forwards to IdP_B. Next, on successful authentication, the IdP_B generates an SSO response token (in a SAML format as a default format) forwarding it to the SP (step 4). At this point of the SSO, the user agent updates the user profile with the new SAML token as signed by IdP_B. This will allow a user profile to dynamically evolve as the user interacts with SPs of different DE's domains.

Figure 7(b) illustrates the case of dynamic token discovery when the same user interacts with a SP of domain C. The SP's trusted IdP_C has a trust relationship with the IdP_B of domain B, but has no (direct) trust with the IdP of domain A. Now, since the user has updated its profile with the identity token of the last SSO interaction, the same can discover that it has an identity token signed by IdP_B of domain B. Since the token is in a SAML format the user can directly provide it for an authentication with IdP_C. After a successful authentication the IdP of domain C issues a new SAML token to the SP in response to the SSO authentication process. Again, the new token information is stored in the user profile for a subsequent usage.

In case the SAML token from the scenario in Figure 7(a) is expired at a time of a next user interaction, the user profile, storing the IdP_B location of service authentication and (optionally) what token was used for authentication, the user can, first, request a token transformation (step 3') to IdP_B (by obtaining the list of trusted IdPs and) presenting the identity token issued by IdP_A. Second, on successful authentication, the user presents the new identity token to IdP_C of domain C (step 3).

The above scenarios can be generalized to an N-step authentication process where a user starts with the list of trusted IdPs of a given IdP and continues with the respective lists of trusted IdPs for any IdP in the main list, thus forming a graph (Web) of trusted IdPs (we included an optional link to IdP's respective list of trusted IdPs for each IdP definition, refer to Figure 6). In this case, an algorithm for finding a matching token is a breadth first search algorithm with no loops.

VI. Identity Management Standards

In a distributed environment, users access in one session services located on different administrative domains and need to be authenticated by each of them. If users would have to sign in each time a different domain is accessed and to remember and manage all the different security credentials, the system will not be scalable and become almost impossible to use with a big number of players. In order to allow users to sign in just once and then access services on other domains (single sign-on), organizations establish trust relations between them (on a contractual basis) and allow access to their resources to users which have been authenticated by one of their trusted partners. This is know as *identity federation* and many specifications and implementations are dedicated to it.

Identity federation means sharing of identity information between domains which have a trust relationship or agreement. Once a federation is established, users can experience single sign-on (SSO) inside the circle of trust. SAML and Liberty Alliance define standards for federating identities and single sign-on (SSO).

SAML [16], developed by OASIS, is an XML-based framework for communicating user authentication, authorization and attribute information. SAML provides XML formats and protocols for encoding and exchanging identity information. SAML assertions allow principals to make statements about a subject's authentication, attribute, or authorization details. A subject is uniquely referred to by using an Identifier which can be a real name or a pseudonym. SAML focuses on authentication and attribute statements while authorization statements are the focus of XACML [27]. SAML assertions provide a good way of exchanging authentication information between parties using different and incompatible authentication technologies. Because of this, we are going to use SAML in our model to achieve interoperability between different standards.

SAML also provides standards for federation creation and SSO. However, though SAML v2.0 is very flexible and offers many choices, in practice it is yet hard to establish identity federations with it [6]. Some of the reported reasons are listed below:

1) Long deployment times. For example, deploying SAML-based projects can take weeks or even months with a single partner. One reason for that is the lack of standardized mechanisms for meta-data exchange and trust establishment.
2) Administrators need to familiarize themselves with the details of SAML v2.0 and have a deep understanding of the way federations are secured.
3) SAML 2 has many choices (for profiles and bindings, attributes and identifiers etc.) but lacks guidance on what is the most appropriate to choose.
4) The implementations available today require administrators to provide answers to fundamental questions that require deep insight into the SAML 2 standard: how to manage trust between providers and metadata describing them,which SAML profiles and bindings to use, which messages and what part of each message should be signed, which identifiers and attributes should be exchanged and how, etc.
5) Administrators need to establish point-to-point federation connections with each new partners. This connections take time and affect the scalability of the system when moving from just a few partners to hundreds or thousands.
6) In order to allow small organizations with fewer resources and technically unsophisticated administrators to deploy these standards, the implementation should be easy to deploy and to configure.

To overcome the above shortcomings, Ping Identity[6] and their partners have been working on developing dynamic SAML [6] which should minimize the steps administrators must perform to configure SAML connections securely.

Liberty Alliance provides open SAML based standards for federated network identity. The most relevant technology specifications developed by the Alliance are Identity Federation Framework (ID-FF) [7] and Web Services Framework (ID-WSF) [8]. As of the new SAML version (v2.0) the OASIS technical committee has unified the Liberty standards within one SAML identity framework with a rich set of identity profiles.

Liberty ID-FF defines identity federation as the linking of distinct user's accounts at the Service Provider and Identity Provider sites. The account linking (or identity federation) is done with the user's consent and must be audited. Liberty ID-FF defines the following required steps for setting up a federation:

1) First of all, businesses form circles of trust based on Liberty architecture and operational agreements that define trust relationships between them.
2) Users federate the isolated local accounts they have with the businesses from the circle of trust. When this happens, the local identifiers (e.g. usernames) of the user are not exchanged between the sites, but instead they exchange opaque user handles.

After this, the users can experience SSO and login at the IdP site and then gain access to the SP sites federated with the IdP. The user needs to allow introductions such that sites of the federation can discover when the user recently accessed a site in the circle of trust and ask the user to federate the accounts. The user can also find a link to trusted SPs from a web site of the IdP. Liberty Alliance specifications are difficult to understand and use for mainly the same reasons we mentioned in the above subsection for SAML. Although business could benefit form deploying Liberty Alliance identity federation solution, the standard is too heavy and organizations face implementation hurdles.

Moreover, it is not always easy for users to discover which accounts they can federate or for SPs to discover which IdP a user is using. This is the case in bigger circles of trust with several IdPs. Liberty ID-FF specifies an optional introduction profile based on cookies which could potentially solve this problem. The idea is to set up a common domain for the circles of trust and to use a common domain cookie accessible by all parties (user, SPs, IdPs). This solution has many shortcomings because it relies on cookies and because common domains need to be updated when trust relations change.

WS-Trust [25] and WS-Federation [23] define standards for federating identities by allowing and brokering trust of identities, attributes and authentication between participating Web services. WS-Trust defines a service model called the Security Token Service (STS), and a protocol for requesting and issuing security tokens. The kind of tokens that a Web Service accepts are described using WS-SecurityPolicy. WS-Federation defines federation as a collection of domains that have established relationships for securely sharing resources. WS-Federation builds on the STS service of WS-Trust and provides mechanisms that simplify interactions between users, IdP (or STS) and SPs. WS-Federation allows to determine policies for obtaining services and cross organizational identity mapping.

[6]Ping Identity Corporation http://www.pingidentity.com

OpenID[7] is a decentralized framework for digital identity. The underlying idea is that users can identify themselves on the web like Web sites do with URIs. OpenID allows a user name/password login. The user name is the personal URI and the password is safely stored on the OpenID Provider. To login to an OpenID-enabled Web site, the user is required the OpenID URI and then gets redirected to the OpenID Provider to authenticate. After authentication, the OpenID Provider sends back the user to the web site with the required identity information to login.

CardSpace[8] [1] is an identity selector for Microsoft Windows. It allows users to have different identities, each represented by a card. When a users needs to authenticate to a web site or a web service, CardSpace pops up a set of suitable information cards for the user to choose from. Each card has some identity data associated with it, though not stored actually in the card. The cards can be issued either by an Identity Provider or by the user himself (self-signed).

The CardSpace model is close to our user profile functionality with the difference of having static updates and with no additional information for a token transformation service. However, CardSpace is a suitable underlying technology for a user-centric profile management.

VII. Conclusion

We have presented an identity management model targeting identity interoperability for DEs. The model bridges main identity standards by using SAML as a unified message-level protocol for querying and obtaining authentication assertions. By using SAML one can automate the process of identifying entities in a distributed environment. We adopted the use of a user-centric profile to keep an abstract view of user's available identity information such as identity certificates, user name/passwords, public/private keys, etc. The user profile is replicated and encrypted on trusted peers.

We presented the core interoperability model, its architecture and message flow. Then, we presented the extension of the model to service compositions. To scale to service composition, we adopted the use of proxy certificates with two main policy settings: limiting a service scope and a level of aggregation. The extended model provides the end-user with the ability to control the use of its identity information in case of service aggregations.

Acknowledgements

Hristo Koshutanski was supported by the Marie Curie EIF iAccess (#038978) fellowship of the 6th Framework Program of the European Commission. Mihaela Ion and Luigi Telesca were supported by the project EU-INFSO-IST ONE (#034744) of the 6th Framework Program of the European Commission.

[7] http://openid.net

[8] http://www.microsoft.com/net/cardspace.aspx

References

[1] CardSpace documentation and resources. http://msdn2.microsoft.com/en-us/netframework/aa663320.aspx.

[2] OASIS Extensible Resource Identifier (XRI). http://www.oasis-open.org/committees/xri.

[3] Rdf primer - turtle version. http://www.w3.org/2007/02/turtle/primer/.

[4] P. Ferronato. Architecture for digital ecosystems, beyond service oriented architecture. In *Proceedings of the 1st IEEE Conference on Digital EcoSystems and Technologies (DEST'07)*, 2007.

[5] P. Ferronato. *Digital Business Ecosystems*, chapter Ecosystem oriented architecture (EOA) vs SOA. European Commission, 2007. http://www.digital-ecosystems.org/book/de-book2007.html.

[6] Patrick Harding, Leif Johansson, and Nate Klingenstein. Dynamic security assertion markup language. simplifying single sign-on. *Security & Privacy, IEEE*, 6(2):83–85, March-April 2008.

[7] ID-FF. Liberty Identity Federation Framework (ID-FF), 2007. http://www.projectliberty.org/resources/specifications.php.

[8] ID-WSF. Liberty Identity Web Services Framework (ID-WSF), 2007. http://www.projectliberty.org/resources/specifications.php.

[9] M. Ion, A. Danzi, H. Koshutanski, and L. Telesca. A peer-to-peer multidimensional trust model for digital ecosystems. In *Proceedings of IEEE International Conference on Business Ecosystems and Technologies (IEEE-DEST'08)*. IEEE press, February 2008.

[10] M. Ion, H. Koshutanski, V. Hoyer, and L. Telesca. Rating agencies interoperation for peer-to-peer online transactions. In *In proceedings of the 2nd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'08)*, pages 173–178. IEEE Computer Society, 2008.

[11] Kerberos. The kerberos network authentication service (v5), 2005. IETF RFC 4120.

[12] H. Koshutanski, M. Ion, and L. Telesca. A distributed identity management model for digital ecosystems. In *Proceedings of the 1st International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'07)*, Valencia, Spain, October 2007. IEEE press.

[13] T. Loukopoulos and I. Ahmad. Static and adaptive distributed data replication using genetic algorithms. *Journal of Parallel and Distributed Computing*, 64(11):1270–1285, 2004.

[14] F. Nachira, P. Dini, A. Nicolai, M. Le Louarn, and L. Rivera Leon, editors. *Digital Business Ecosystems*. European Commission, 2007. http://www.digital-ecosystems.org/book/de-book2007.html.

[15] Francesco Nachira, Paolo Dini, and Andrea Nicolai. *Digital Business Ecosystems*, chapter A Network of Digital Business Ecosystems for Europe: Roots, Processes and Perspectives. European Commission, 2007. http://www.digital-ecosystems.org/book/de-book2007.html.

[16] SAML. Security Assertion Markup Language (SAML), 2005. http://www.oasis-open.org/committees/security.

[17] SPKI. SPKI certificate theory, 1999. IETF RFC 2693.

[18] S. Tuecke, V. Welch, D. Engert, L. Perlman, and M. Thompson. RFC3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, 2004. http://www.ietf.org/rfc/rfc3820.txt.

[19] W3C. RDF Primer, W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[20] W3C. RDF/XML Syntax Specification, W3C Recommendation, 2004. http://www.w3.org/TR/rdf-syntax-grammar/.

[21] W3C. Representing vCard Objects in RDF/XML, 2004. http://www.w3.org/TR/vcard-rdf.

[22] O. Wolfson, S. Jajodia, and Y. Huang. An adaptive data replication algorithm. *ACM Transactions on Database Systems*, 22(2):255–314, 1997.

[23] WS-Federation. Web Services Federation Language (WS-Federation), 2006. http://www-106.ibm.com/developerworks/webservices/library/ws-fed.

[24] WS-Policy. Web Services Policy Framework (WS-Policy), 2004. http://www-106.ibm.com/developerworks/library/specification/ws-polfram.

[25] WS-Trust. Web Services Trust Language (WS-Trust), 2005. http://www-106.ibm.com/developerworks/library/specification/ws-trust.

[26] X.509. The directory: Public-key and attribute certificate frameworks, 2005. ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.

[27] XACML. eXtensible Access Control Markup Language (XACML), 2005. http://www.oasis-open.org/committees/xacml.

Preliminary 2009 Conference Schedule

http://www.iaria.org/conferences.html

NetWare 2009: June 14-19, 2009 - Athens, Greece

- SENSORCOMM 2009, The Third International Conference on Sensor Technologies and Applications
- SECURWARE 2009, The Third International Conference on Emerging Security Information, Systems and Technologies
- MESH 2009, The Second International Conference on Advances in Mesh Networks
- AFIN 2009, The First International Conference on Advances in Future Internet
- DEPEND 2009, The Second International Conference on Dependability

NexComm 2009: July 19-24, 2009 - Colmar, France

- CTRQ 2009, The Second International Conference on Communication Theory, Reliability, and Quality of Service
- ICDT 2009, The Fourth International Conference on Digital Telecommunications
- SPACOMM 2009, The First International Conference on Advances in Satellite and Space Communications
- MMEDIA 2009, The First International Conferences on Advances in Multimedia

InfoWare 2009: August 25-31, 2009 – Cannes, French Riviera, France

- ICCGI 2009, The Fourth International Multi-Conference on Computing in the Global Information Technology
- ICWMC 2009, The Fifth International Conference on Wireless and Mobile Communications
- INTERNET 2009, The First International Conference on Evolving Internet

SoftNet 2009: September 20-25, 2009 - Porto, Portugal

- ICSEA 2009, The Fourth International Conference on Software Engineering Advances
  - SEDES 2009: Simpósio para Estudantes de Doutoramento em Engenharia de Software
- ICSNC 2009, The Fourth International Conference on Systems and Networks Communications
- CENTRIC 2009, The Second International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services
- VALID 2009, The First International Conference on Advances in System Testing and Validation Lifecycle
- SIMUL 2009, The First International Conference on Advances in System Simulation

NexTech 2009: October 11-16, 2009 - Sliema, Malta

- UBICOMM 2009, The Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies
- ADVCOMP 2009, The Third International Conference on Advanced Engineering Computing and Applications in Sciences
- CENICS 2009, The Second International Conference on Advances in Circuits, Electronics and Micro-electronics
- AP2PS 2009, The First International Conference on Advances in P2P Systems
- EMERGING 2009, The First International Conference on Emerging Network Intelligence
- SEMAPRO 2009, The Third International Conference on Advances in Semantic Processing