A role-based architecture for seamless identity management and effective task separation

Evangelos Kotsovinos[1], Ingo Friese[2], Martin Kurze[2], and Jörg Heuer[1]

[1] Deutsche Telekom Laboratories
[2] T-Systems
{firstname.lastname}@telekom.de

Abstract. Today's on-line end user experience is compromised by the need for managing multiple redundant identities for access to various services — such as email accounts, in order to ensure a clear separation of tasks that users perform in different capacities. Approaches based on Single Sign On (SSO) have focused on the provision of interoperability and trust management solutions required to allow users to log in once and use multiple on-line services. In this paper, we argue that Single Sign On provides neither adequate privacy preservation nor sufficient fine-grained separation of tasks, as it requires that a user performs all tasks — whether e.g. personal or professional — using the same identity. We propose Identity and Role Management (IRM), a new approach to identity management, combining the benefits of SSO and user-centric frameworks: it allows a user to be authenticated as conveniently as with SSO, to still achieve an effective separation of tasks she performs in different capacities through the use of different roles, and to retain full control of her private and sensitive data. Additionally, it facilitates fine-grained service customisation, supporting a personalised on-line experience. Our experiments with real users demonstrate the effectiveness, transparency, and user acceptance of our solution.

1 Introduction

Today's on-line end user's experience is hampered by the complexity of managing identities. For instance, users often resort to maintaining multiple email identities, established with different authorities, in order to achieve a clear separation of emails they send in different capacities or what we term *roles* (such as professional email versus personal email)[3]. As another example, users are typically required to maintain a separate login/password pair for every web site they wish to register with.

[3] It is worth noting that the use of the term "role" differs from the one commonly found in other role-based systems: for us a role is one of the many sub-identities a given user may have, whereas in RBAC it denotes a class of users with common characteristics.

Please use the following format when citing this chapter:

Kotsovinos, E., Friese, I., Kurze, M. and Heuer, J., 2007, in IFIP International Federation for Information Processing, Volume 232, New Approaches for Security, Privacy and Trust in Complex Environments, eds. Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R., (Boston: Springer), pp. 289–300.

Developments in the field of Identity Management (IDM) focus on supporting Single Sign On (SSO) — the facility that allows users to log in once and access a wide range of on-line services. This is achieved by ensuring the interoperability of the various on-line accounts and by forming *federated trust relationships* to allow on-line services to delegate user authentication to reputable third-party authorities. However, while SSO removes the need for maintaining multiple identities, it does little to facilitate a clear separation of roles and to provide adequate privacy protection. Users have to maintain a single identity and use that for all their on-line interactions, whether they are professional, personal, or in any other capacity.

In this paper we propose *Identity and Role Management* (IRM), a scheme based on *roles* as a means to achieve separation of the different capacities in which a given identity can be used. Our approach is shown to combine the benefits of multiple identities and SSO; it separates tasks as effectively as multiple identities, and provides the convenience of SSO to the users.

Furthermore, our system allows users to have an *adaptable level of control* of their private data, based on their individual requirements and preferences — in full compliance with current demands for user consent. Similarly to user-centric approaches — discussed in more detail in Section 6.2, our system allows privacy-sensitive users to retain full control of their personal details. At the same time, it enables convenience-seeking users to outsource their attribute and identity storage and management to trusted third parties.

The rest of this document is structured as follows: Section 2 discuss the shortcomings of existing approaches. Section 3 describes roles as an enhancement of identity management. Section 4 presents our implementation, the experiments undertaken with real users, and the results obtained. Section 5 discusses open issues of our framework and outlines future work. Section 6 positions our work in the context of related work, and Section 7 concludes.

2 Background

The *Single Sign On* concept envisages users logging in only once, for example on a web page of an on-line service, and visiting further services or web-based applications without the need to log in again. The user can thus experience an unhindered, seamless usage of services. The key concept behind Single Sign On is federation, denoting the establishment of common references between accounts or identities in different repositories or services. Microsoft Passport[4] as well as several other systems have been developed based on this concept [16].

For services to exchange information about the user, or authenticate a user for the other service respectively, these services need to have established a trust relationship with each other. So, if a given service B trusts a given service C, users of B could be authenticated by C. In that case, C is called

[4] `http://www.passport.com`

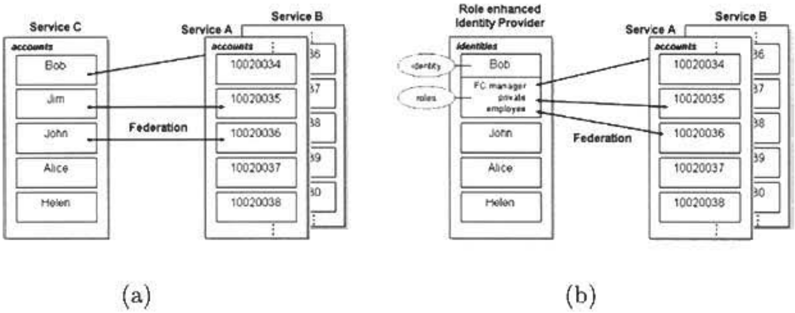


Fig. 1. Identity management using a) conventional Single Sign On, and b) role-enhanced Identity Provider

an *Identity Provider* (IDP) — as shown in Figure 1(a). Longer chains of trust relationships may be established, for instance if another service C trusts service B to authenticate users, and service B trusts service A in turn to authenticate users. The concept of service federation has been described in the Security Assertion Mark-up Language specifications (SAML [13, 14]).

While SSO represents a significant progress in the way user authentication and identity management are handled over the conventional approach, we believe it provides neither adequate *separation* of tasks that users perform in different capacities nor sufficient *privacy protection*. Even using SSO, users need more than one identity to separate, for instance, private from professional email accounts, as shown in Figure 1(a): SSO associates an account with an identity, and as all accounts can be associated with the same identity this causes linkability, which compromises privacy [17]. Additionally, user data needs to be exchanged between federated services, which may not be trusted by the user for doing so. This relates to the concept of user-centric identity management, discussed in Section 6.2.

3 Framework

3.1 Overview

We propose Identity and Role Management (IRM) to enhance existing identity management approaches. IRM is based on augmenting identity management with the concept of a *role*. This is not to be confused with the meaning that the term role has in access control; here, it refers to the capacity in which a given user performs a certain action — for instance, "private", "employee of a company X", "soccer club manager".

In conventional SSO systems — as shown in Figure 1(a), a user's identity is associated with one account in each service the user is registered with. Our approach allows associating roles, not entire identities, with accounts, allowing

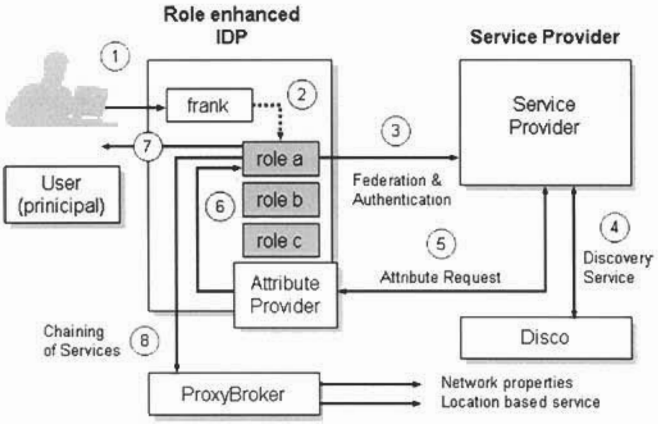


Fig. 2. Framework architecture and operation

a user to have multiple accounts with a single service in different capacities, and achieving effective task separation.

An example is shown in Figure 1(b), where a single user, Bob, is able to maintain two separate accounts in service A, each one through the use of a different role — "private" and "employee". This allows the user to achieve a clear separation of the tasks he performs using different roles. Additionally, Bob can maintain an account with service B through his "soccer club manager" role, facilitating sign-on using a single identity without compromising task separation.

When Bob reads his email using his mobile phone on the company premises he is automatically assigned the "employee" role, as long as he does not explicitly request not to, and data charges are billed directly to the company's account. When the same user chooses the role "soccer club manager", the charges are billed to his club account. Furthermore, the network is configured to provide all relevant club contacts to his device, which then displays them in his phone book application. Accordingly, the network provides the user's family contacts, when he uses the "private" role.

3.2 IRM architecture

The main parts of our IRM framework are shown in Figure 2. The core component is a trusted *Identity Provider* (IDP) module, which handles user accounts and is enhanced with role capabilities. The Identity Provider is responsible for providing authentication credentials based on the current role that a user has. It is important to note that individual services are relieved of handling user authentication themselves, as they can use the Identity Provider as an authentication service in the same way the would use a SAML enabled IDP.

A user is free to specify herself as her own IDP, if so desired, in order to maximise local control of her identity data and management.

Our role enhancement of the trusted IDP is combined with an *Attribute Provider* (AP). When a service requires information about a certain attribute of a user — such as name and email address, it submits a request to the AP, which holds the reference between the current role and the appropriate attribute. A set of user attributes can be seen as a profile that represents a role. Thus, role and profile are related concepts, closely linked to the same principal — the role referring to a user's identity and the profile referring to a user's attributes. For privacy, each user is free to select an Attribute Provider of her choice; indeed, if so desired, a user may specify herself as her provider, thus ensuring that personal details are held locally, for privacy.

The *Discovery Service* is a component that supports locating a service instance that holds certain attributes for a user with a given identifier. Upon request, the Discovery Service returns a resource offering-an endpoint and credential — for a given web service provider. The Discovery Service can be used to locate not only third-party services, but also IRM architecture services, such as the IDP and AP. This facilitates the dynamic discovery of the AP that can provide information about a certain user attribute, enabling the distribution of user attributes among different APs.

In IRM, the retrieval of user related attributes is not limited to Web Service instances, as is typically the case with a conventional attribute provider. The ProxyBroker is able to retrieve information gathered from other domains, such as the network and user properties.

IRM can be used both in a setting of transient federation — a non-permanent federation relationship, as defined in SAML 2.0 — and in a permanent federation case.

3.3 Authentication Using IRM

Let us consider a user Frank, who attempts to access a resource of a service provider, an on-line shop. Frank does not have a current log-on session on this site and is unknown to the service. The service provider sends an HTTP redirect to the role-enhanced Identity Provider. The HTTP redirect contains a SAML `<AuthnRequest>` requesting that the Identity Provider provides an assertion about the requesting user. The request asks that the Identity Provider sends back an identifier. Until this step the process is similar to those described in SAML 2.0.

From this point on, the operation of IRM in order to authenticate a user encompasses the following steps, as shown in Figure 2:

1. The user will be *challenged* by the IRM to provide valid credentials. The user provides valid credentials and identifies himself as Frank. The IRM looks up user Frank in its IDP and finds references to the various roles that Frank has created in the past.

2. The user is prompted to *choose the role*, either manually or with the help of a context-aware application framework — such as a context middleware system [2]. In our example, Frank chooses his role, named "private shopper", which he uses for shopping online. A security and session context is created for the user. The IRM creates a *name identifier* to be used for this federation, which is linked to Frank's role.
3. The IRM *redirects* the user back to the service that requested authentication. The service validates the digital signature of the SAML response and the SAML assertion. The provided name identifier is used to create a session context for Frank in his role. Frank is authenticated now through the "private shopper" role, and can be referenced via the corresponding name identifier.

3.4 Service Customization Using IRM

The on-line shopping service that Frank uses is capable of providing personalized book recommendations. In order to provide effective recommendations, the service wishes to acquire information about certain attributes of users by communication with the IRM. Additionally, information such as a user's address can be used to simplify the ordering and delivery process of goods, without requiring that the user types in the address repetitively.

For a service provider to obtain information about user attributes in the IRM framework, the following steps are taken (as shown in Figure 2):

4. The service provider requests the *discovery* of a web service instance that holds attributes for a user with a given identifier. The discovery service provides one or more references for that service — such as URLs — and credentials with which the service provider can access the service at that endpoint on behalf of the end user.
5. The service provider *requests the user attribute* in question from the Identity Provider by submitting the user identifier. The Identity Provider then maps the identifier to the appropriate user and role, and provides the value of the requested attribute. The value itself can be retrieved from a number of sources:
   - The Identity Provider's internal attribute list that is linked to the current role of the user.
   - Through direct interaction with the user, in case the attribute in question is not available through an AP, or is part of data or user information that is considered sensitive or personal.
   - From other services that are chained via interfaces, proxies or a broker — for instance, if the attribute in question is the current location the user

In our example, Frank's personal literature preferences (mystery novels) are retrieved directly from the attribute list of the "private shopper" role he has chosen. When Frank connects to the on-line bookstore using his "professional" role, the book recommendations he will be given will be related to new technology, relevant to his subject of work.

For the low-level mechanisms to facilitate all the previous steps, we use off-the-shelf protocols that are defined in standards and drafts of the Liberty Alliance. Such concepts include identity service discovery, permission-based attribute sharing, interaction service, and service chaining. This is important to allow interoperability, compatibility, and extensibility of our framework.

4 Implementation and observations

We set up a few sample services — such as an on-line shop and a messaging service based on Jabber[5] on Sun and Apache web servers, simulating the conditions of a heterogeneous platform. Two back-end Identity Providers were run in virtual machines on the same server, a Sun Fire V440 with four 1.28GHz UltraSPARC III CPUs, 8GB RAM, and 200GB SCSI HDD. Sun Access Manager[6] and RSA Security's Access Manager[7] were used for access control. These were connected to the role management extension component we implemented, which handles roles and attributes. The user interface of our system for managing roles was part of the front-end component, which communicated with the back-end over Java RMI, and run on a 3.4GHz P4 with 2GB RAM, 250GB IDE HDD, and an NVIDIA Quadro FX1400 card.

4.1 Experiments.

We conducted experiments with 36 real users to evaluate our system in three dimensions: effectiveness, robustness, and acceptance by users. Our experiments in the above setting have demonstrated that our prototype has been fully operational, successfully handling role-based identities as described in the previous sections in all tested cases. We plan to undertake further performance and scalability experiments in the future. In terms of integration challenges, we observed that in several cases existing identity management systems are not fully conformant with the Liberty Alliance's standards, and this non-conformance is not always adequately documented.

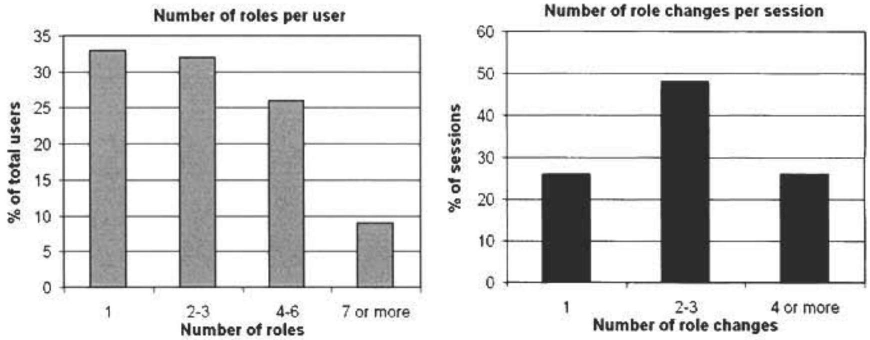
4.2 User tests.

We asked 36 users to use the sample services, while their identities were managed by our system, and describe their impressions. Initially a web-based login and password scheme was used to allow users to enter, switch, and manage

[5] `http://www.jabber.org/`

[6] `http://www.sun.com/software/products/access_mgr/`

[7] `http://www.rsasecurity.com/accessmanager`



(a) Roles per user

(b) Role changes per session

Fig. 3. Usage of identity and role management

roles. We measured the number of roles they opted to use (with "general user", "buyer", and "employee" being the most frequently used ones), and the number of role changes in each session. These are shown in Figure 4.1.

Users liked the role-based approach but — not surprisingly — did not want to actively change their role through an additional role-selection interface. Furthermore, they prefer not to have to select their role in advance of accessing the service. They would prefer engaging in the login process after the intended service asks for it — e.g. when the on-line bookstore browsed requests their details in the last ordering steps, after their basket has been filled. Finally, users wish to be immediately aware of the currently active role at any time, without having to request this information from our framework.

4.3 Refined interfaces and further tests.

Based on the above feedback, we developed a next version of interfaces for our prototype, which comprises a an easy-to-use role-selector that enables the following features. Firstly, it increases *awareness* of the current role by changing the desktop background every time the currently active role changes. Secondly, it communicates role changes to the user through intuitive *3D transition* metaphors, for instance by showing different roles' desktops on the different surfaces of a cube — implemented using a modified version of Project Looking Glass[8] v0.6.2. This also allows the user to work on several desktops simultaneously, each for one role. Thirdly, it reduces the manual overhead of role transition by enabling *speech-based* role selection and management.

Our users felt significantly more comfortable with the new interface. They were highly aware of their current role and were happy to be disburdened from manually managing their current login status — "who am I right now in this application?". In addition, they liked the concept of carrying their role-based set of attributes with them from application to application, still separating the individual roles from each other.

[8] `http://www.sun.com/software/looking_glass/`

5 Discussion

This document has presented our first steps towards devising a comprehensive IRM framework. However, there are a number of social and technical challenges that need to be overcome for IRM to realize its full potential.

Adoption

Although a full-scale switch to IRM would require that users gradually change familiar means of authentication, we believe that their current *frustration* with identity multiplicity and their *privacy concerns* will act as significant incentives for doing so. Additionally, our framework is built to operate in a partial deployment setting, to accommodate for its gradual adoption. Furthermore, we have successfully conducted initial user studies and are currently in the process of performing more, in order to understand and further improve the usability properties — and thus the adoption potential — of our framework.

New Requirements to User Interfaces

While we wish to provide for an abundance of roles per user, adequate to cover the different capacities in which she performs on-line activities, we also wish to *not overwhelm the user with the management of roles.* New types of user interfaces will be required to allow handling and switching between roles without placing administration burden on the user. The use of contextual information for automatic role inference could be one technique to be investigated in this area. Furthermore, the multitude and heterogeneity of devices from which a user connect to IRM places additional interface adaptation requirements.

Compatibility with Existing Services

To ensure the faster adoption of IRM and reduce the corresponding barrier to entry, we have designed our architecture to support *backward compatibility* with existing services and *interoperability* with established IDM standards. Additionally, IRM has been designed to be operational in a *partial deployment.* When an IRM-enabled user interacts with a non-IRM-enabled service that manages authentication and accounts on its own, the role of the user presented by the Identity Provider is seen as the identity of the user by the role-agnostic service.

Privacy as a Design Principle

IRM implements the following mechanisms for reassuring users about the protection of their private data. Firstly, IRM *prevents* the *association* of data referring to different roles by services and other third parties. An on-line user in different capacities is represented as two different users on the service provider side, and only the trusted IDP is able to trace back the roles to an identity. Additionally, IRM allows a user to retain *full control of her private data* by specifying herself as her Attribute Provider — or even an IDP, thus ensuring her privacy. Finally, IRM has been developed in accordance to privacy protection standards and legislation.

6 Related Work

The main areas of research related to our work are the Liberty Alliance identity specifications, the user-centric community, and (role- and attribute-based) access control.

6.1 Liberty Identity Service Interface Specification

Realizing the importance of moving towards a more fine-grained separation of on-line tasks that users perform, the Liberty Alliance has devised the Liberty Identity Service Interface specification (ID-SIS) [9] . This provides an XML schema for describing user profiles and attributes in a structured manner, and recommends a set of interfaces for querying providers of such profiles to obtain user attributes.

The approach we propose in this document is orthogonal to ID-SIS. We propose the use of roles as a key mechanism for achieving separation of tasks and privacy preservation. Furthermore, we present a comprehensive architecture for deploying IRM in an on-line setting, including facilities for IRM-based authentication, role assignment and management, and service customization. Within IRM, the ID-SIS specification can be employed as a common scheme for describing user attributes.

6.2 User-centric community

Driven by the users' growing privacy concerns regarding the handling of their authentication information, user-centric identity management approaches such as CardSpace[9], Yadis [11], SXIP [18, 6], and Persona [19] have gained popularity. These go beyond the Liberty Alliance's standards and federation concepts to allow individual users to retain full control over their own identity management, without requiring the presence of a provider of an external provider of identification information. Essentially each user manages — and is liable for — its own provider of identification information.

However, despite the thrust behind such systems at the time of writing, we believe that there are technical challenges that need to be addressed. In most such systems, it is not clear how identities can be securely *ported* between devices to allow a user to authenticate from different terminals. Additionally, protecting identities on the user side from unauthorised human users — for instance other members of the same household — needs to be done in a

[9] http://msdn.microsoft.com/webservices/infocard/

passwordless way. Finally, incorporating single sign-on to such systems is not trivial.

As described before, our system can support user-centric identity management functionality by registering the user herself as her attribute provider. This allows full, local control of her properties and sensitive personal data, while at the same time retaining the advantages of provider-assisted identity management such as simple Single Sign On mechanisms and ease of use.

6.3 Role- and attribute-based access control

Ferraiolo and Kuhn [4] provided an early formal description of role definition and membership for RBAC. [12] administers roles, role relationships, and access rights. [10] defines roles as sets of rights and duties. [7, 8, 5] combine roles and policies for applying RBAC to open, large-scale systems. [3] specifies positive and negative security policies associated with roles, as well as role inheritance. Attribute-based access control [1, 8] makes fine-grained access control decisions based on user attributes and their combinations. RBAC has been implemented in web-based enterprise environments [15].

Our work draws inspiration from role- and attribute-based access control systems, but at the same time is complementary to them. We focus not on the mechanisms to control which user groups have access to a given on-line resource, but rather on how such systems can be interlinked to provide an unhindered on-line experience for the user, separation of tasks, and privacy protection. We plan to evaluate the possibility of employing an off-the-shelf RBAC solution for access control on individual resources.

7 Conclusions and future work

The on-line behavior and requirements of users indicate the need for a facility to allow using a single digital identity in different capacities, thus retaining the benefits of Single Sign On while not compromising the separation of tasks achieved using multiple on-line identities. We proposed Identity and Role Management (IRM), enhancing traditional identity management approaches by introducing roles as a powerful mechanism to achieve a clean separation of tasks performed by a user in different capacities. Furthermore, we presented an architecture for implementing and deploying the IRM framework. Additionally, we described how our framework supports adaptable local control of private data and attributes, facilitating user-centric privacy preservation. Also, through experiments with real users, we demonstrated the effectiveness, transparency, and acceptance of our solution.

We believe that IRM represents a natural next step in the area of identity management, enabling the convenient use of services, ensuring fine-grained separation of tasks, protecting user privacy, and reducing the amount of authentication data that has to be administrated on the service side. Also, roles enhance personalization, by allowing the network to customize the provided services based on the current user role. Ultimately, IRM represents a step towards decreasing the barrier of using new services, by simplifying the overall on-line user experience. In the future we plan to investigate ways to make IRM intuitive and near-transparent to the users, bootstrapping issues of our identity management platform, role federation for service customisation, and role instantiation to allow a more flexible association of roles with identities.

References

1. P. Bonatti and P. Samarati. A unified framework for regulating access and information release on the web. *Journal of Comp. Sec.*, 10(3), 2002.
2. N. H. Cohen, J. Black, P. Castro, M. Ebling, B. Leiba, A. Misra, and W. Segmuller. Building Context-Aware Applications with Context Weaver. Research Report RC 23388, IBM, Oct. 2004.
3. N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The Ponder Policy Specification Language. In *Proc. of the Policy2001 Workshop*, Jan. 2001.
4. D. Ferraiolo and R. Kuhn. Role-Based Access Controls. In *Proc. of the 15th NIST-NCSC Conf.*, 1992.
5. R. J. Hayton, J. M. Bacon, and K. Moody. Access Control in an Open Distributed Environment. In *Proc. of the IEEE Symp. on Sec. and Priv.*, 1998.
6. J. Merrells. DIX: Digital Identity Exchange Protocol, Mar. 2006.
7. D. Jonscher and K. R. Dittrich. Argos – A Configurable Access Control System for Interoperable Environments. In *DB Sec., IX: Status and Prospects*, 1996.
8. N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a Role-Based Trust Management Framework. In *Proc. of the IEEE Symp. on Sec. and Priv.*, 2002.
9. Liberty Alliance Project. Liberty ID-SIS Personal Profile Service Spec., 2003.
10. E. C. Lupu, D. A. Marriott, M. S. Sloman, and N. Yialelis. A Policy Based Role Framework for Access Control. In *Proc. of the 1st ACM RBAC '96.*
11. J. Miller. Yadis Specification, Version 1.0, Mar. 2006.
12. M. Nyanchama and S. Osborn. Access Rights Administration in Role-Based Security Systems. In *Proc. of the 8th IFIP WG 11.3 Working Conf. on DB Sec.*, volume A-60. Elsevier, Aug. 1995.
13. Organization for the Advancement of Structured Information Standards (OASIS). Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML), Apr. 2002.
14. Organization for the Advancement of Structured Information Standards (OASIS). Security Assertion Markup Language (SAML) V2.0 Technical Overview, Sept. 2005.
15. J. S. Park, R. Sandhu, and G.-J. Ahn. Role-based access control on the web. *ACM Trans. Inf. Syst. Sec.*, 4(1), 2001.
16. A. Pashalidis and C. Mitchell. A taxonomy of single sign-on systems. In *Proc. of the 8th Australasian Conf. in Inf. Sec. and Priv.*, July 2003.
17. A. Pfitzmann and M. Hansen. Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management — A Consolidated Proposal for Terminology. Research report, TU-Dresden, May 2006.
18. SXIP Networks. The SXIP 2.0 Overview, Mar. 2006.
19. K. Toth and M.Subramanium. Requirements for the persona concept. In *Proc. of RHAS'03*, Sept. 2003.