Security and Privacy Enablers for Future Identity Management Systems

Marc BARISCH[1], Elena TORROGLOSA Garcia[2], Mario LISCHKA[3],
Rodolphe MARQUES[4], Ronald MARX[5], Alfredo MATOS[4],
Alejandro PEREZ Mendez[2], Dirk SCHEUERMANN[5],
[1]*University of Stuttgart, Institute of Communication Networks and Computer Engineering, Pfaffenwaldring 47, 70569 Stuttgart, Germany,*
Tel: +49 (711)685-60217, Fax: +49(711) 685-50217, Email: barisch@ikr.uni-stuttgart.de
[2]*University of Murcia, Murcia, Spain;* [3]*NEC Laboratories Europe, Heidelberg, Germany*
[4]*Instituto de Telecomunicações de Aveiro,Universidade de Aveiro, Aveiro, Portugal*
[5]*Fraunhofer Institute for Secure Information Technology (SIT), Darmstadt, Germany*

Abstract: In recent years, Identity Management (IdM) has gained a lot of attention in industry, standardisation and academia. In particular, a couple of research projects, like Daidalos or Prime, have invested considerable effort to bring IdM forward, to take advantage of features like improved usability and security. Nevertheless, there are important issues that have not been addressed so far. The SWIFT project leverages IdM as a key technology of the Future Internet, tackling problems like the integration of the network and application layer from an IdM perspective as well as the use of electronic identity cards. Moreover, aspects like the integration of several user devices, backward compatibility and a new access control infrastructure are required by future IdM solutions. We consider all these aspects by extending existing IdM solutions with six new security and privacy enablers that are part of the overall SWIFT framework. These enablers have been partially implemented towards a new IdM architecture. First evaluation results of the implementation are promising to pave the way towards future IdM solutions.

Keywords: Identity Management, Privacy Protection, Virtual Identity

1. Introduction

Identity Management (IdM) is becoming more and more important for telecommunication operators as well as for service providers (SP) in the Web 2.0 area. It is expected that the IdM market will enormously grow in the future [1]. IdM solves a couple of problems that users face today. Among these problems are identity fragmentation, i.e. a user has to manage many different accounts with various SPs. Hereby, each account typically requires its own username and password combination, which has direct consequences on the usability and security. Since users tend to use weak passwords or to reuse username/password combinations across different providers, severe security threads are imposed. Moreover, manual authentication against each SP is required, which also means that each user account needs to be filled with the needed user attributes leading to a decreased usability. IdM provides features like Single Sign-On (SSO), Single Log-Out or Attribute Provisioning and improves not only the user experience by allowing seamless service usage but also security.

A couple of IdM solutions, like Shibboleth [2] that is mainly used in the academic sector or OpenId [3] and Windows CardSpace [4] for the Web 2.0 area, exist and are underway to be widely introduced. This is an important step to improve usability and security within the Internet. Nevertheless, a detailed gap analysis [5] showed that these

solutions have a couple of shortcomings, which need to be addressed in the future. Among the shortcomings is the non-existing integration of network and application layer, access control infrastructures that do not consider the high degree of distribution of participants, or insufficient privacy support for users.

The Daidalos project already started to extend IdM towards the network and introduced the VID concept [6] to increase user's privacy. Based on the pioneer work of Daidalos, the SWIFT (Secure Widespread Identities for Federated Telecommunications) project continued to enhance IdM solutions targeted on telecommunication operators. As a result an extended IdM architecture has been designed that contains a couple of security and privacy enablers addressing requirements of future IdM solutions. In particular we provide a cross-layer privacy solution that enhances existing work and a new access control infrastructure. Moreover, we employ electronic ID cards, provide an integrated view on user devices and tackle the problem of backward compatibility with existing solutions.

The remainder of this paper is structured as follows. In Section 2, we introduce the relevant IdM parts of the SWIFT architecture. Afterwards, Section 3 introduces the SWIFT security enablers in detail, followed by the planned evaluation in Section 4. Section 5 concludes this paper.

2. SWIFT Architecture

The SWIFT project aims to provide an IdM solution that overcomes shortcomings of existing IdM solutions. We took a cross-layer approach that integrates network and applications layer from an IdM perspectives. That means we allow the user to consume services via different networks, by establishing an integrated view on the user identity across services and networks. At the same time user's privacy is of uttermost importance and needs to be preserved.

Based on a gap analysis [5], which was focused on a privacy preserving identity view across services and networks, we identified shortcomings of existing IdM solutions and came up with a set of requirements for future IdM systems. These requirements (c.f. Section 2.1) indicated that a new role definition is necessary, which is introduced in Section 2.2. The new role definition serves as a basis for the design of the security and privacy architecture in Section 2.3.

2.1 Requirements

We extracted the most important requirements from [5] and present them in the following.

- **Req. 1 - Overcome Identity Fragmentation:** User's digital identity is fragmented. Thus user attributes are distributed across various accounts with different SP. The users have to be supported to manage this highly distributed information by means of a unified view across systems and providers.
- **Req. 2 - Cross-Layer IdM**: Most IdM solutions target SSO for application layer services, neglecting the network layer with inconvenient and even dangerous consequences. In order to achieve cross-layer IdM, network authentication must be compatible with application layer authentication. That means we need an IdM solution that takes application layer as well as network layer into account.
- **Req. 3 - Improved privacy features**: Privacy preservation is one of the most important properties of IdM for user acceptability. The considerations of current research on privacy-enhancing technologies [6] need to take network properties into account, because network identifiers can be used for correlation.
- **Req. 4 - Support for multiple devices**: Current IdM solutions do not take into account that an end user owns more than one device and uses these devices to consume services. By providing an integrated view across all end user devices, taking into account the diversity of devices as well as of identities, the usability and security of IdM can be further increased.

- **Req. 5 - No dependency on online components**: Many IdM solutions depend on components like Identity or Attribute Providers in order to work. That means these systems need 100% availability, which is difficult to guarantee. Moreover, if a user has no network connectivity, the system should still work for limited period of time. Therefore, solutions are needed that work temporarily without dependencies on online components.
- **Req. 6 - Backward compatibility**: It is not reasonable to build new IdM solutions that do not interwork with already existing solutions. Therefore, new IdM solutions have to be either compatible with already existing systems or have to provide opportunities to interwork with those legacy systems.

2.2 Roles

As a consequence of the above introduced requirements, we came to the conclusion that the classical role definition of IdM systems, as presented below, is not sufficient. The classical role definition identifies the main actors in an IdM system, their functions and how they interact. Although each classical IdM solution defines its own roles, there exists a list of main roles that appears in most of them, sometimes with different names but with almost the same description and behaviour:

- **User**. The User (sometimes referred as End User, Client, Subscriber or even Device) is the role that owns the identity information. He wants to benefit from the IdM system by outsourcing the management of his identity information and accessing the different services that are available by means of this information.
- **Service Provider (SP)**. The SP (aka Relying Party) consumes the User's identity information. This information is used, on the one hand to ensure that the User is the one he claims to be, and on the other hand to determine if the User is authorized to obtain the service. Additionally, the identity information can be used to customize the service.
- **Identity Provider (IdP)**. The IdP is the role that stores User's information and provides it to different SPs upon request but only if the User has authorized that. Its main tasks are to manage User's identity information, to authenticate the User and to release identity information to SPs.

Since we came to the conclusion that this role definition is not sufficient, we came up with a couple of modifications. In particular, we started to subdivide the role of the IdP into three different roles:

- **Attribute Provider (AttP).** The AttP is a specialization of the traditional IdP role, which only takes the user identity information management part of the functionality. The information is defined as attributes, i.e. pairs of *name* and *value*.
- **Authentication Provider (AuthNP).** The AuthNP is also a specialization of the traditional IdP role. It only assumes the responsibility of the User's authentication.
- **Identity Aggregator (IdAgg)**. The IdAgg manages virtual identities, which are defined as the aggregation of identity information (credentials and attributes) from different providers. The IdAgg creates a new level on the identity management hierarchy, placing itself between the SPs and the AttPs and AuthNPs [7].

The main difference between the SWIFT roles and the traditional ones is that they allow disaggregating the different functionalities from the traditional IdPs. This allows the introduction new levels (e.g. IdAgg) in the IdM hierarchy (c.f. Figure 1). Virtual identities provide a unified view over identity information, fulfilling Req. 1. The introduction of the IdAgg addresses also Req. 2, since it allows decoupling identity information from the concrete AttP, and so from the concrete authentication technologies.

2.3 Architecture Overview

Based on the requirements, we have designed the in Figure 1 depicted architecture that is part of the overall SWIFT IdM architecture [10]. It contains five security enablers on the user device interconnected by the VID Manager and the Credential Manager. Some enablers require support by the IdAgg. The sixth enabler, the Distributed Management

Enabler (DPME), is distributed across the IdAgg, AttP, AuthNP and SP. With these enablers we are in the position to provide new opportunities with respect to security, privacy and usability.

The end user interacts with the system by the *VIDManager* (VIDM), which is the central component for all IdM related tasks on the user device. It provides a graphical user interface in order to select, create or destroy virtual identities, establishes sessions with SPs and IdAggs, and makes use of the connected security enablers. Moreover, it allows the configuration of attribute release policies, i.e. which SP is entitled to access which user attribute. If necessary, it triggers the creation of credentials with the Credential Manager.

The *Credential Manager* (CM) is responsible for the creation of credentials that are necessary for the authentication and authorization against the SP and the IdAgg. Within the SWIFT context these credentials are also known as framework statements [2]. The Credential Manager provides an abstract interface to the Electronic ID Card (EIDC), the Credential Bootstrapping Enabler (CBS), and Anonymous Credential Enabler (ACE).

The *EIDC* provides among others a secure storage for credentials including user attributes. This not only increases the user device security, but also enables the end user to consume services independent of connections to the IdAgg or AttP (Req. 5). In addition, the *ACE* can provide an additional level of privacy (Req. 3) by employing the mechanism of anonymous credentials [6]. If needed, the CM can create specific credentials to interoperate with other IdM systems by means of the *CBS*, i.e. provide backward compatibility (Req. 6).

Since a user can have very different devices with respect to their resource and security capabilities that are used in different contexts (e.g. business vs. private), the *Identity Transfer Enabler (ITE)* enables the usage of identities across devices (Req. 4). Moreover, the VIDM is in the position to control the representation of the user identity on the network and on the service layer by means of the *Cross-Layer Pseudonym Manager* (Req. 2).

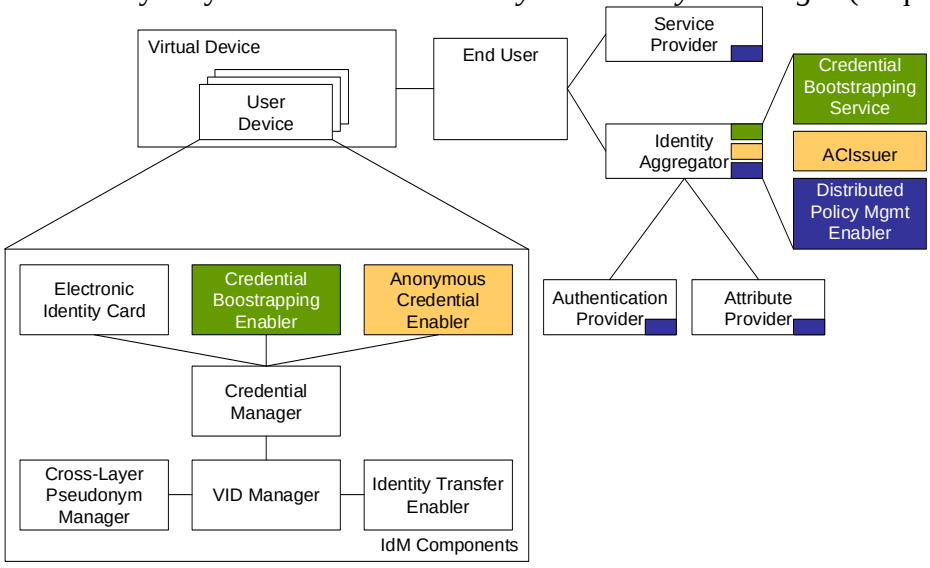

Figure 1: SWIFT Architecture

3. Security and Privacy Enablers

The above introduced architecture gave an overview on the basic functionality and the interrelationships of the enablers. In the following we go into the details of the enablers.

3.1 Anonymous Credential Enabler

Current IdM systems often use X.509 certificates to communicate end user related information to an SP imposing additional privacy threats. First, the certificate might reveal

more personal information than necessary. Second, transactions can be linked based on the same certificate in different contexts.

This enabler aims to put the user in control of the disclosure of his personal information by means of anonymous credentials (AC) and thus enhance the user's privacy to support the principle of minimal data disclosure (Req. 3). AC are cryptographic tokens that allow users to prove statements about themselves and their relationships with organizations, anonymously. In particular, we employ cryptographic mechanisms like non-iterative zero-knowledge proofs, and advanced signatures and encryption schemes [8].

The end user employs the IdAgg to create, manage and aggregate his ACs. Relying on privacy-dedicated cryptography fundaments of the AC concepts, the end user (i.e. through his IdAgg) is then able to prove that an asserted attribute statement is true without revealing any further information than those derivable from the statement.

The SWIFT architecture benefits from the security and privacy features provided by digital ACs in four ways. First, ACs feature unlinkability of transactions performed by the same user (i.e. by the use of pseudonyms). Second, the user can perform zero-knowledge-based attribute certification to reveal as less information as possible. Third, an AC cannot be forged, even if multiple non-authorized entities collude. Forth, ACs support the revocation of end user's privacy in cases of malicious end users.

AC support requires the introduction of an AC Issuer (ACIssuer) function responsible for AC generation, performed within the IdAgg. The process of anonymity/pseudonymity revocation is done by a neutral trusted authority, the AnonResolver. The AnonResolver can be deployed as separate trusted entity, which is offline until asked by an SP that is testifying illegal behaviour by an end user. In such cases, the AnonResolver must evaluate pre-defined conditions to reveal the malicious end user's identity. The AC enabler depends on the availability of the IdAgg, i.e. Req. 5 is not fulfilled.

3.2 Electronic Identity Card

Within the SWIFT IdM framework, the SP needs to retrieve authentication statements and user attributes regarding a virtual identity from the IdAgg. This presents two main drawbacks that limit the scalability of the system. First, the IdAgg may become a bottleneck for authentication and attribute releasing, since it is defined as an intermediate point for the transmission of the virtual identity information. Second, the IdAgg is a single point of failure, i.e. if the IdAgg is not available, the access to a SP would be impossible.

To solve these problems (Req. 5), the end user is equipped with an EIDC, issued by the IdAgg, The EIDC provides a subset of the IdAgg functions needed for immediate service access, allowing the end user to consume services provided by SPs by utilizing the EIDC in conjunction with his terminal without contacting the IdAgg. This minimizes the dependability of online components (Req. 5) and improves the aspect of user centricity keeping the user in control of his personal data (Req. 3).

Current smart cards provide a lot of valuable security features for SWIFT. This comprises PIN verification as well as cryptographic operations (including key generation, encryption and signature generation) and enables to protect access to data stored on the EIDC. We use the card to store virtual identifiers and credentials (including user attributes) and for the generation of authentication and attribute statements [7]. Since the data on the EIDC and the data stored by the IdAgg/AttP might diverge, we have to synchronise in regular intervals. This additional feature avoids the need for re-issuing a new card if e.g. attribute values are changed and need to be updated, or if new services are introduced. The synchronisation interval between the EIDC and the IdAgg has to be selected in a way that data on the EIDC is up to date and that at the same time the dependency on the availabity of the IdAgg is minimized.

3.3 Credential Bootstrapping Enabler

The SWIFT IdM framework uses extended credentials [7], which are essential to support the new role concept combined with the VID concept. That means on a first glance that all participants must be capable to support the extended credential format. However, we must be aware that there might be SPs that do not adhere to the new credential format or systems that employ particular security technologies out of legacy reasons. Since we cannot exclude those SPs and systems out of usability reasons, we need provide interworking possibilities (Req 6).

This is achieved by the CBS. It allows bootstrapping SP/system specific credentials based on a previously established trust relationship between the IdAgg and the non-SWIFT SP/system. The trust relationship is either established in a bilateral way for example by the exchange of X.509 certificates or by means of a PKI. Policies describe which kinds of tokens are exchanged between the IdAgg and non-SWIFT SPs/systems in dependency of the trust level that has been previously configured. At the same time we still benefit from the SWIFT specific properties like SSO, unlinkability, pseudonymity, etc.

The CBS is an additional function provided by the IdAgg as shown in Figure 2. If a user wants to consume non-SWIFT services it can trigger a CBS credential request at the IdAgg. The CBS credential request contains among other parameters the type of the needed credential. If the user is not already authenticated with the IdAgg, it needs to authenticate in order to obtain the credential. With the provided credential the user is in the position to consume the service.

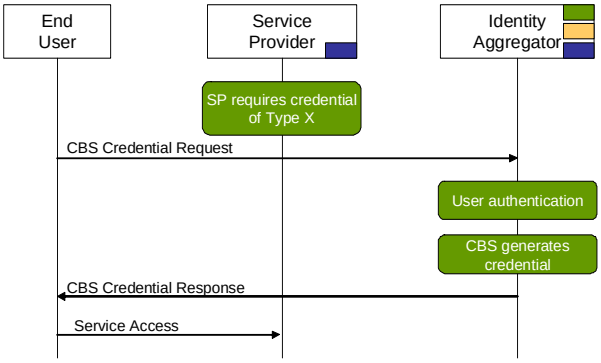

Figure 2: Credential bootstrapping service

3.4 Identity Transfer Enabler

The Identity Transfer Enabler (ITE) targets the problem of providing the user with a unified view across all his user devices, making consumed services as independent from a particular device as possible. This is achieved through the Virtual Device concept (c.f. Figure 1) [9] that provides security associations and discovery mechanisms between the user's devices. Each device has its particular advantages with respect to resources or security features. For example, a mobile phone has strong authentication capabilities based on a SIM card, whereas a notebook has a large display. The ITE makes it possible that the notebook benefits from the mobile phone with respect to authentication, i.e. it can trigger the generation of credentials to access services that have strong authentication needs, e.g. imposed by the SP through authentication contexts [9].

This does not mean that all virtual identities a user owns, can be used on all devices. It must be possible that a certain subset of the user's identities with a higher security level than others, e.g. all business identities, can only be used on a subset of his devices, e.g. business devices. The ITE exploits meta data about the virtual identities and meta data about the devices to achieve this. The meta data about identities describes the contexts

(private, business, etc.) in which these identities can be used and which authentication methods are supported by a particular identity. In contrast meta data about a device gives information about the security properties of the device, among them is the available security hardware as well as the installed software versions, and the usage context of the device. Basically we can differentiate three categories of identities: Directly usable identities, Indirectly usable identities and Unusable identities. Directly usable identities can be used directly on the device without interaction with other devices. That means the device has the required context and fulfils the requirements of the authentication procedure with respect to the needed protocols and algorithms. On the other hand, indirectly usable identities have the correct usage context, but depend with respect to authentication to one of the devices owned by the user. E.g. the needed authentication protocols are not supported. Finally, some identities cannot be used at all, since the usage context of the identity and device do not fit.

Policies evaluate the different kinds of meta data and restrict the usage of identities. The policies are either system-wide policies that are specified by a system administrator or user-specific policies specified by the user. In the case of user-specific policies, a user has the right to overrule the policies, whereas system-wide policies should be strictly enforced.

The ITE in combination with the Virtual Device concept provides the user with a seamless usage experience across all his devices. At the same time it does not negatively affect the security, since it considers usage contexts and security properties of devices.

3.5 Cross-Layer Privacy Enabler

IdM systems put a strong focus on protecting the user's privacy when interacting with services. The SWIFT IdM Framework [10] employs pseudonyms between the IdAgg and SP [7], based on SAML [11], in order to protect every interaction of the user.

However, privacy threats do not only exist at the application layer. The network stack employs its own identifiers that can be used to link the user's pseudonyms and thus compromise the high level mechanisms employed by the IdM framework. This problem can only be mitigated by following a cross layer approach where the instantiation of a pseudonym from the IdM layer would affect the way that the network stack is used.

The SWIFT approach to provide cross layer privacy support to the framework consists on using Virtual Network Stacks (VNS) [12], controlled on-demand by the IdM system through the VIDM (c.f. Sec 2). Each VNS is represented by its own virtual network interface directly linked to a Virtual Identity, creating different network addresses at every layer (e.g. MAC Address, IP Address), thus disguising the user under several layers of pseudonyms. Since the virtual identifiers (at different layers) are used on a per identity basis, correlation between different virtual identities is mitigated, allowing a user to use multiple VIDs in the same terminal without the risk of them being correlated.

The major contribution of *VIDM* in the Swift framework is the management of each VNS, and related resources, providing the bridge between the network and the application layer, enforcing privacy protecting policies that determine how a VNS is required depending on application needs and SP interaction.

3.6 Distributed Policy Management Enabler

As shown in Figure 1 the DPME is deployed at various entities related to the management of the user's identity and the related attributes. Such a distributed deployment is necessary to tackle multilateral security and privacy aspects. In the SWIFT project we identified various aspects which have to be considered in the authorization policies such as privacy of attributes, identity management, resource access, delegation of privileges and agreements (see [13] for details). As these aspects have to be coordinated in a cross layer and cross

domain approach, current standards like XACML with their monolithic approach are not feasible. It is highly unlikely that all the aspects could be integrated into one policy at one single entity. Instead the SP should incorporate decisions of other entities (e.g. those of the IdAgg and AttP) into its own rules.

Therefore, we developed XADML (eXtensible Authorization Deduction Markup Language) as an extension to the existing XACML standard to incorporate decisions of other entities (called *authorization domains*) into local ones. This approach has several advantages. It provides a new abstraction layer which hides the details of the policies at other authorization domains, thus allow an independent modification as well as confidentiality of the own decision rules and the related attributes. As these distributed policy requests have to support bridging of different application areas, the related subject, object and action of a request might change when send to another authorization domain. XADML provides the incorporation of decision as well as attributes from remote entities on the level of policy sets, incorporating the merging of obligation from distributed entities. The required extensions to the language and the architecture have been presented in [14] and represent an essential part of the overall SWIFT architecture.

4. Evaluation

The evaluation of the in Section 3 introduced enablers has recently started and will be documented in [15]. Since each enabler was subject to different design goals, we need to apply different evaluation methodologies. Table 1 provides an overview on the evaluation methodology with the corresponding evaluation metrics applied for each enabler.

Prototypical implementation is the main evaluation methodology. It proves the feasibility of the designed enablers and allows for the quantification and measurement of enabler-specific performance metrics as indicated in Table 1.

Table 1: Evaluation of security and privacy enablers

Enabler	**Design goal**	**Evaluation Methodology**	**Performance Metrics**
Anonymous Credential Enabler	Improve user privacy	Formal security verification	n.a.
Electronic Identity Card	Mitigate dependencies on online components	Prototypical implementation	• Implementation complexity • Synchronisation effort between IdAgg and EIDC
Credential Bootstrapping Enabler	Interworking with non-SWIFT IdM systems	Not planned	n.a.
Identity Transfer Enabler	Improve security and usability	Prototypical implementation	• Reduction of signalling effort • Number of authentication procedures • Implementation complexity
Cross-Layer Privacy Enabler	Improve user privacy	Prototypical implementation	• Time to setup VNS • Delay introduced by VNS
Distributed Policy Management Enabler	Support for highly distributed policy decisions	Prototypical implementation	• Policy Decision Time • Implementation complexity

5. Conclusions

When exploring the potential of the presented IdM solution, it is possible to identify that providing a cross-layer approach is not a straightforward process, especially considering the complex security and privacy interactions with network and services. We have outlined a

set of requirements that highlight the need to support dynamic and highly distributed environments that characterise the Future Internet, where multiple accounts, devices, and diversified scenarios are common. Such constellations raise the bar for security and privacy solutions. These requirements are solved by several security and privacy enablers, that are consistently integrated in a cross-layer architecture.

In this paper we described the SWIFT approach that required an in-depth look at the network and the application layer to create a cross layer IdM solution. The proposed security and privacy enablers fill in the conceptual gaps of IdM current systems that lack cross layer solutions and pave the path towards future IdM solutions. This is in fact the major contribution of the proposed approach, which considers the network and services as a whole, powered by a vertical (across layers) and horizontal (across providers) aggregating identity concept that takes into account not one, but all aspects of Future Internet solutions, and based on this provide appropriate concepts for security and privacy.

The SWIFT project continues to explore the results presented above, especially considering the implementation of several demonstrators that clearly shown the described enablers, as part of the roadmap to further study and evaluate the SWIFT cross-layer IdM concepts.

Acknowledgement

This work was supported in part by the European Union under the FP7 programme (SWIFT project).

References

[1] Neuenschwander, M., et al.:VantagePoint 2007: Trends in Identity Management, Burton Group 2007
[2] Shibboleth Architecture Technical Overview, Working Draft 02, June 2005.
[3] OpenID Authentication 2.0 -Final, Dec. 2007, http://openid.net/specs/openid-authentication-2_0.html
[4] Bertocci, V. et al.: Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities, Addision-Wesley Longman, 2008
[5] Matos, A. (ed.): Gap Analysis and Architecture Requirements , SWIFT Deliverable 202, 2008
[6] Sarma, A., et al.: Virtual Identity Framework for Telecom Infrastructures, Springer Wireless Personal Communications, Special Issue on “International Mobile Telecommunications – Advanced, 2008.
[6] Camenisch, Jan et al.: Design and implementation of the idemix anonymous credential system,: Proceedings of the 9th ACM CCS, 2002 pages 21- 30, Washington, DC, USA
[7] López, G, et al.: A SWIFT Take on Identity, Computer, IEEE Computer Society, 2009, Volume 42, Pages 58-65
[8] Blum, M. et al.: Non-interactive zero-knowledge and its applications, STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing, 1988, pages 103--112, Chicago
[9] Marx, R. (ed.): Specification of General Identity-centric Security Model that supports user control of privacy, SWIFT Deliverable 302, 2009
[10] Girao, J. (ed.): First Draft of the Identity-driven Architecture and Identity Framework, SWIFT Deliverable 203, 2008
[11] Cantor, S. et al.: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard saml-core-2.0-os, March 2005.
[12] Matos, A. et al.: “Preserving privacy in mobile envinronments”, IEEE Globecom 2007, Washington D.C., USA
[13] Lischka, M. et. al: Towards Standardization of Distributed Access Control, W3C Workshop on Access Control Application Scenarios, 17./18. November 2009, Luxembourg
[14] Lischka, M. et. al: Deductive Policies with XACML. 2009 ACM Workshop on Secure Web Services, Chicago, Illinois, USA, November 2009
[15] Lutz, D. (ed.): Simulation, Modelling and Prototypes, SWIFT Deliverable 504, 2010 (to appear)