



Cross Layer Privacy Support for Identity Management

Rodolphe MARQUES, Rui FERREIRA, Alfredo MATOS

Instituto Telecomunicações de Aveiro, Campus Univ. Santiago, Aveiro, 3810-193, Portugal

Tel: + 351 234 377900, Fax: + 351 234 377900

Email: rmarques/rferreira/alfredo.matos@av.it.pt

Abstract: One of the most important objectives of Identity Management (IdM) Systems is to provide end user privacy. However, these concepts rarely extend beyond the application layer. In the IST SWIFT project a special attention is given to cross-layer Identity Management support, and in this paper we show why applying only IdM solutions is insufficient to preserve user privacy if network mechanisms are not considered. We present a solution to retain user privacy by using network pseudonyms closely coordinated with the IdM framework proposed by the SWIFT project. We include these concepts in the IdM framework and present the necessary architecture and functional mechanisms required to support the privacy extensions.

Keywords: Identity, Privacy, Network, Cross-layer support

1. Introduction

As computer network permeate our daily life, privacy becomes a crucial requirement for the adoption of pervasive technologies, and consequently, for the Future Internet. By focusing on key aspects such as security and privacy, Identity Management (IdM) is shaping into an important part of future systems and architectures. Particularly, the strong authentication and authorization framework it provides, aimed at retaining user privacy, is an important feature. IdM Systems focus on privacy aspects that deal with the interaction between users and services. In this context the SWIFT IdM Framework [1] is a SAML [2] based approach that employs pseudonyms between the End User (EU) and Service Providers (SP), providing privacy to the user by avoiding correlation across different SPs. Through minimum disclosure policies and technologies for user attributes, this approach protects every user interaction by hiding or withholding sensitive information.

However, privacy threats are not limited to the application layer, which is the main scope of IdM solutions. Many threats stem from the network stack, where different properties in the communication channels used between EU and SP can be explored to undermine the privacy provided by the IdM framework. Network protocols and identifiers can be used to link information on upper layers as noted in [3]. Therefore, we must prevent that IdM related pseudonyms (SAML pseudonyms) can be linked through network stack identifiers. This privacy threat, detailed in Sec. 3, must be thwarted in order to support a true cross layer privacy solution.

In this paper we take advantage of a cross layer pseudonym solution, namely Virtual Network Stacks [3], to enable a full privacy solution in IdM scenarios. The key challenge, addressed in Sec. 4, is the integration of the network pseudonym solution and SWIFT IdM system. The remainder of the paper is organized as follows: Sec. 2 presents an overview of the base SWIFT IdM system; Sec. 3 presents the network threats to IdM; Sec. 5 presents the architecture and use cases for the proposed solution. We finish the paper in Sec. 6 by discussing and summarizing the proposed concepts and future work.

2. Overview of SWIFT IdM Architecture

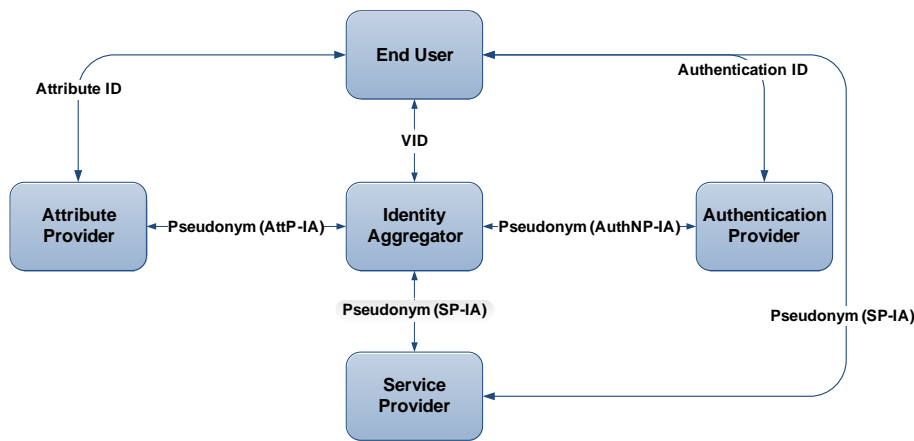


Figure 1: SWIFT Architecture Overview.

The SWIFT architecture attempts to cover a wide range of issues by employing a cross layer identity solution. With a SAML framework at the heart, SWIFT uses IdM concepts to tackle network based interactions, well beyond SP interactions. We focus on the relevant aspects that can be used to drive cross layer privacy, addressing issues that range from the network layer to the application layer, providing an approach to ensure the user privacy.

Some of the most important features provided by the SAML framework are the pseudonyms established between the SP and the EU, which protect the user identity, making the user anonymous at the SP. Stemming from the SAML approach, Single Sign-On (SSO) also proves of key value for the end user enabling a more secure and controlled environment, through the usage of adequate statements that proves that a user is authenticated and controls a specific identity. Of key importance is also the use of Virtual Identities (VID), which means that the user, through the framework only discloses the relevant parts of his identity, appearing as different virtual identities exist, instead of a single user. This increases the overall privacy of the user.

The SWIFT architecture defines that the SAML framework does not only apply to service interaction, but can be used at different levels in the network, establishing a consistent cross-layer approach to security and privacy. It is composed of five main functional elements, as seen in Figure 1. The End User (EU) plays a central role in the SWIFT architecture. As the owner of the identity information, the EU selects appropriate policies and mechanisms that handle how information is disclosed within the system. By using several identities, the EU is able to subscribe/consume services made available by Service Providers (SP), which in the SWIFT scope is anything that provides additional value to an EU, including network services. The SP consumes EU's identity information in the form of Authentication Statements in order to ensure that the EU is really who he claims to be, and Attribute Statements in order to ensure that the EU meets all the necessary authorisation requirements to consume the services. However, the Authentication Provider (AuthNP) is the element responsible for the actual EU authentication. It creates the necessary authentications statements that also act as SSO Tokens, and are used by the EU in order to prove that he is the rightful owner of a given VID, thus proving their authenticity to SPs and network services.

All the aforementioned processes occur through the Identity Aggregator (IdAgg) that acts as the coordinating element of all the EU identity related information. While not necessarily the information holder of all EU related information (e.g. attributes, credentials), the IdAgg is able to retrieve it, most of which lies with Attribute Providers (AttP). Also, by interacting on the EU's behalf, and employing different pseudonyms to each peer entity within the framework, the IdAgg greatly enhances EU privacy. AttPs are

the elements responsible for storing EU's attributes. One IdAgg keeps track of the EU's AttP and fetches the necessary attributes needed by the EU to consume a service at the SPs. The SWIFT framework employs a minimum disclosure policy when it comes to disclosing EU's attributes to SPs.

Stepping back, we acknowledge that SWIFT provides a user centric framework where the EU has control over its identity. It protects the user information against unauthorized access and exercises control on information access, by disclosing only what is absolutely necessary for the EU to consume a service. It also uses pseudonyms in the communication channel between all the entities, being the IdAgg the only element that can link EU's VIDs. However, the network can jeopardize the aforementioned privacy efforts by presenting several threats as discussed in the following sections.

3. IdM Privacy Threats on the Network

IdM stresses the use of pseudonyms for privacy purposes, to prevent correlation of user information. However, by inspecting network traffic and packet data, it is possible to see the identifiers that flow in the communication, and link the identities behind them through different techniques [3, 4]. The identifiers that appear together in the same packet serve different purposes, which are aggregated into four relevant layers: Link, Network, Transport and Application. Each of these identifiers offers unique information: Link Layer Identifiers, such as the MAC address, uniquely identifying each terminal (and potentially the user behind it) in the network; the Network Layer Identifiers, IP addresses, offer topological positioning and subject the user to tracking and location threats; the Transport Layer Identifiers are usually reused network layer identifiers coupled with port information, inherit the network layer properties; finally, the Application Layer Identifiers do not share a unique definition and vary on each application, but are commonly use to uniquely represent the users, i.e. email addresses (*name@domain*).

3.1 Linking SAML Pseudonyms

As a part of IdM systems, SAML supports the use of pseudonyms that ensure a user can perform multiple uncorrelated interactions with the same service provider. While guarantying that correlation at the SP is impossible using the mentioned pseudonyms alone, it also clearly states that “correlation may be possible through non-SAML handles” [5].

Maintaining different pseudonyms for a certain layer will not ensure privacy, if through vertical linkage we can create a relationship between a lower layer identifier with a higher layer identifier, thus inferring that those pseudonyms belong to the same user. In practice, a user may present many SAML pseudonyms to the same SP, but if all interactions are made using the same IP address the SP could infer that all actions were performed by the same terminal (and user) defeating the purpose of the pseudonyms. This issue does not stem from SAML, but rather from the fact that SAML interactions are carried atop identifiers over which SAML has no control.

The aforementioned correlation event will always occur in two situations on the SAML framework: upon the first authentication of the user against the IdAgg, and when contacting an SP. This procedure maps well to an IdAgg-initiated authentication scenario: when the EU is already authenticated with the IdAgg, and wishes to access an SP, the EU asks the IdAgg to issue an Authentication Statement for the SP. This Authentication Statement is now bound to a SAML pseudonym (created during the initial enrolment/subscription) used by the SP to identify the EU. Assuming that the EU has more than one subscription with the same SP it would be easy for the SP to, through vertical linkage, infer that those pseudonyms belong to the same user. We consider the set of identifying information that SAML accredits to one identity (one pseudonym) and expand it to include the identifiers

from the lower layers. With this expansion we can align the pseudonymity features from SAML with equivalents in the network stack.

3.2 Giving out identifiers before identity is chosen

In SAML based interactions, the user chooses his identity, represented by the employed pseudonyms, when accessing a SP. Before authentication takes place SAML has no considerations regarding pseudonymity considering that all non-authenticated users are equal. This does not hold true for network interactions, because when a terminal sends a message into the network it is immediately disclosing identifiers. For IdM it means that upon the moment a terminal starts sending packets it is asserting an identity, in the form of network identifiers (IP and MAC addresses). This implies that, prior to contacting a service the user's identity must already be instantiated in terms of local identifiers (at the terminal). If not, when the EU contacts an SP it is already presenting a set of network identifiers, without an associated SAML pseudonym. After the initial SP contact, the SP will initiate an EU authentication to determine the user's identity (pseudonym). This is another important use-case defined as SP-initiated authentication [7]. From now on, any pseudonym that the EU presents to the SP can be correlated, through the use of the network stack identifiers that the EU presented to the SP in its initial contact.

The contradiction between the typical SAML authentication and the exposed interaction model dictates two main cases that must be handled: 1) the user has already chosen the identity to be employed (and consequently all associated pseudonyms) or 2) the user has contacted a SP but has not yet selected an identity, and special considerations must be made to circumvent this use-case. .

4. Supporting Cross Layer Pseudonymity

Pseudonymity is a core feature of both SAML and the SWIFT IdM framework. The same concepts can be applied on the network stack to support the cross layer design. To achieve this, it is necessary to generate Layer 2, Layer 3 and Layer 4 pseudonyms, different from the real device identifiers, creating multiple identifiers per layer – one for each identity. This has been demonstrated in [3] by using Virtual Network Stacks (VNS), where virtual interfaces are instantiated per identity, creating a network identifier set based on identity while not compromising upper layer mechanisms. This allows a user to disguise itself under several layers of pseudonyms, avoiding correlation between different virtual identities, and at the same time retains operating system and network semantics, given that each VID is assigned its own virtual interface. The results are cross-layer pseudonyms on an identity basis, leading to a VNS being assigned per identity, making it impossible to use the identifiers to correlate different virtual identities.

To support this paradigm we must introduce a control plane that enables the interaction with IdM. This control plane interacts with applications, which provide information useful for network stack management, covered in [3]. However, the real challenge for the SWIFT framework is to determine when and how to apply a new VNS, either when performing IdM operations through the IdAgg or when connecting to an SP.

4.1 Network Pseudonyms and the IdM Framework

To integrate with the SWIFT (SAML based) architecture, it is necessary to generate different identifiers when an identity wishes to connect to an SP (or set of SPs). Since different SPs see different pseudonyms, it is important that the IdM control layer supports such granularity. It is also important to strike a balance between privacy and performance, established by the IdM framework, given that creating multiple network stacks can come

with a hefty toll due to the network implications: creating multiple pseudonyms (addresses and identifiers) will impact the access technology because the device will now have to take care of multiple ongoing communications, e.g. 802.11 will require multiple associations possibly through different channels; addressing impacts will stem from the fact that the user will now use a number of network identifiers proportional to the number of VIDs in use, which reduces the address space available in the network. Because of this it is important to be able to re-use the same network stack for different operations, while still preserving the user's privacy, and thus saving resources.

For the majority of the operations, such as contacting an SP, different SAML pseudonyms will be used assuming that the identity was already selected and properly authenticated, leaving us in the IdAgg-initiated authentication scenario. The decision to generate or reuse a VNS should come from a policy based approach determined by the IdM, which is already aware of authentication, VID, pseudonym and SP. The most straightforward strategy would be instantiating a network stack for each SP, using a different SAML pseudonym. Alternatively, we could map the instantiation of stacks to the user's notion of virtual identity [6] and allocate stacks under a policy of one stack per user.

There is already a set of information divulged at the application layer (e.g. presenting the user's name at different SP links different pseudonyms), for which the IdM is properly equipped to deal with. By reusing the same principals and privacy policies that guide this process, it is possible to select one VNS for different providers. Such policies should be enforced by the IdM control layer, which handles the proper mapping between the SAML identifiers and the network stack identifiers.

The IdAgg can be considered a special case, given its implicit trust properties on behalf of the user. The initial authentication with the IdAgg requires the use of one VID, i.e. requiring one VNS, and yielding a SSO Token [7] that may be used to contact several other SPs. The following authentications of different VIDs at the same IdAgg can re-use the same VNS because the IdAgg is trusted and capable of mapping different pseudonyms to the same VID, thus saving resources that yield no privacy increasing. However, if supplemental privacy countermeasures can be taken and the user can, instead of using only one VNS for an authentication, create a set of predetermined VNSs, which can be used in a round robin fashion – a pool of IdAgg authentication VNS. While this has no effect towards the IdAgg, it introduces confusion on the connection overthrowing any eavesdroppers on the network, and still saves a fair amount of resources.

4.2 Contacting the Service Provider

Previously, we defined the generic policies that cover creating or reusing a VNS to contact an SP, assuming the user is authenticated and has selected a VID – IdAgg-initiated authentication scenario. However, this only covers the scenario where the identity to use is already clear. When the first contact is to the SP, as defined by the SP-initiated scenario mentioned in Sec. 3, the risk of correlation increases, given that a network stack is required to first contact the SP.

Protecting the user in this scenario requires two approaches to be considered: 1) use a new VNS for such events, preventing correlation but triggering a performance bottleneck or 2) use a VNS pool dedicated to this case, from which a VNS is selected on a round robin basis on each occurrence. As soon as the user selects his identity the appropriate VNS can be used instead.

The later approach needs to be carefully evaluated as there will be a point of correlation (Random VNS -VID VNS), regardless of how minimal. This however, strikes the necessary balance due to the fact that we are reducing the number of stacks required to maintain the unlinkability across different identities. By using a round robin approach there will be a

large period of time until re-usage occurs and linkage is possible. Nevertheless, since this is policy driven, it can always state that a new VNS will be used in such cases, incurring in the performance penalties as mentioned in Sec. 4.1, but safeguarding user privacy, especially if we considered this to be the non-standard case as opposed to first authenticating at the IdAgg.

5. Architecture

The previous sections determined how we can couple a network pseudonymity solution and IdM concepts to enhance user privacy. However, by itself the VNS is only a tool that can be leveraged to prevent linking attacks on the network. The strategies that allow control of the pseudonymity solution are just as important, if not more, than the tool itself. To achieve full privacy protection, there must be a clear definition of how the VNS is used by the IdM framework, and more importantly, we must define the entities that provide such control and interaction, and the functions they must support.

The goal of our cross layer privacy proposal is to extend the instantiation of SAML pseudonyms in the IdM layer to the entire network stack. To do so, we connect the SAML operations that require network resources to the creation or re-usage of a VNS. An example of this process can be a user contacting an SP which will cause a VID to be chosen consequently triggering instantiation of SAML pseudonym. This pseudonym instantiation must be properly conveyed to the cross layer privacy, where a decision must be carried out, based on policies, whether to create a new VNS or re-use an existing VNS. In this section we present the functional elements from which the architecture to support cross layer privacy was built on, and we present a practical example of how this solution maps into a real IdM use case scenario.

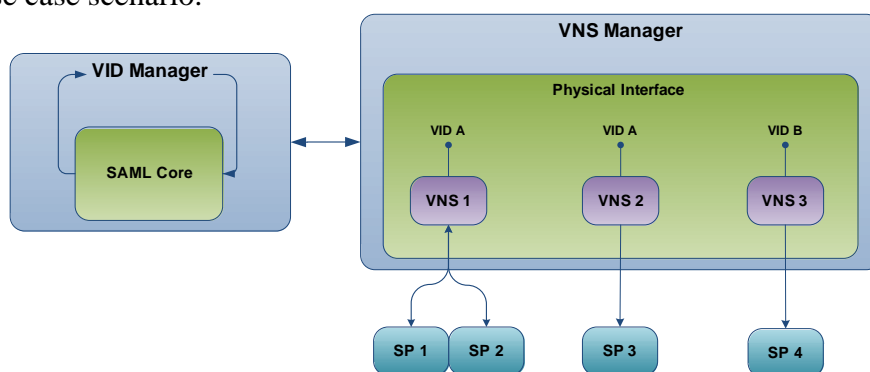


Figure 2: Cross layer pseudonym solution.

5.1 Functional Elements

The functional elements that compose the privacy architecture are built around the principle that there must be a tight integration between the VNS solution and the IdM support functions. This integration is done through the SAML Core which intercepts SAML operations that require network resources for the interaction between a user and a SP. Figure 2 details the cross layer privacy architecture composed by three main components:

- **VID Manager (VID-M):** The VID-M is the central element of this architecture and interacts with all the other elements. It is the interface point between the IdM layer and the VNS functionality and intercepts SAML pseudonym instantiations through the SAML Core. In case network resources are needed it triggers the policy mechanisms that will decide if for a given pseudonym it should use either a new VNS or reuse an existing one.

- **VNS Manager (VNS-M):** The VNS-M element is primary point of interaction with the network stack. It is responsible for the proper usage of the VNS. The VNS-M enables the creation of the necessary network pseudonyms, and configuration of the underlying network stack, providing virtual interfaces for each VNS, simulating the existence of multiple devices in the user's terminal.
- **SAML Core (SAML-C):** The SAML-C element is responsible for bridging the SAML operations performed on the IdM layer that involve to the creation or usage of VIDs, pseudonyms or assertions, to the VID-M. The SAML-C also creates the link to any identity management policies that can come from the IdM layer.

5.2 Operational Overview (Use Cases)

After defining the functional elements of the cross layer privacy architecture, we now describe how these components interact to enable the desired functionality. This operational overview describes the usage of VNSs in two main scenario considered that relate to user authentication, either IdAgg-initiated or SP-initiated (a broader range of use cases, with further details, can be found in [7]). We only present the case of web based services, thus assuming that the EU already has a network connection but it is not consuming any service. Whenever a user attempts to access a (web based) service, the SP will initiate an authentication towards the IdAgg. The identity selection and the VNS interaction must be carefully considered before contacting the SP. This interaction is policy controlled, as discussed in Sec. 4, and can be described in the following three step process:

1. When an EU requires access to a service, he must first select the VID to use, triggering the creation of a new VNS.
2. In case the EU wants simultaneous access to a different SP with the same VID it should use the IdAgg-initiated authentication, given that he is already authenticated and possesses a SSO Token. In this step the EU has two options: 1) use the same VNS (one VNS per VID scenario), or 2) create a new VNS in order to prevent correlation of the same VID with the two SPs (one VNS per SP scenario).
3. In case the EU wants to access a new SP with a different VID, step 1 is repeated and a new VNS is created.

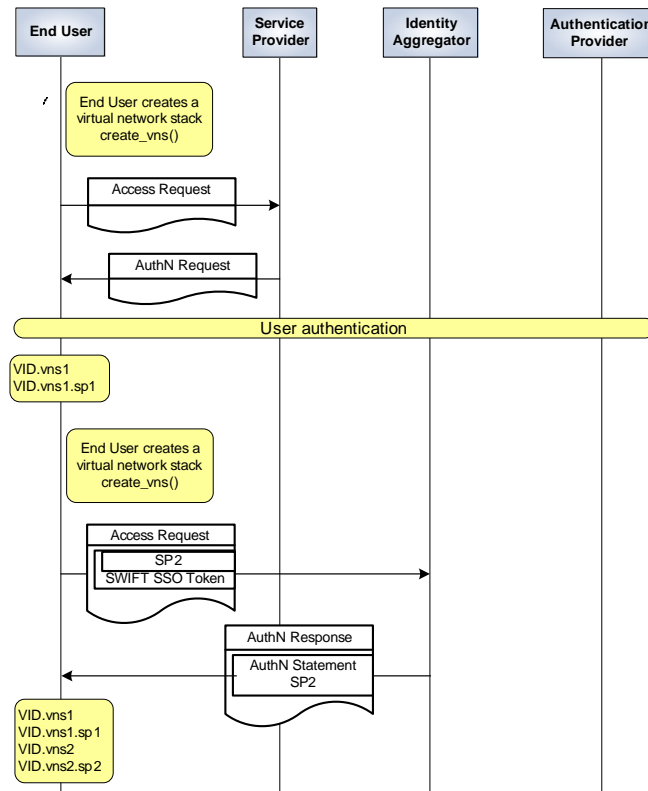


Figure 3: Accessing multiple services using Single-Sign-On mechanisms.

This three step process can be mapped to a detailed interaction shown in Figure 3, which takes place in the network where the EU which to authenticate and consume services. In this interaction the EU starts by selecting a VID, triggering the creation of a new VNS. Afterwards, the EU performs a Service Access operation to consume the service provided by SP 1. While consuming this service, the EU wishes to access SP 2 and due to the fact that he already is authenticated with the IdAgg, only a new VNS is created and used alongside the already existent SSO Token to obtain an Authentication Statement for SP 2. In the end the EU has two VNSs, one per service, making it impossible to link the pseudonyms being used between the EU and SPs, to the same EU terminal. This guarantees that the end user privacy is ensured even whilst consuming services with different VIDs on a single terminal.

6. Conclusion

In this paper we have uncovered a set of conditions that can obliterate the privacy provided by IdM solutions, and how to solve them by using cross-layer pseudonymity solutions on the network stack. The end goal of pushing this approach to the network stack was to prevent any entity in the network from perceiving that the user's multiple identities belong in fact to the same individual. In practise what is achieved by using pseudonyms in the multiple layers of the stack, is that multiple terminals will appear to be in the same network (one per user identity in use) when in fact only one terminal exists. We wanted not only to push the pseudonymity approach to other layers of the network but also to ensure cohesion between them, when using multiple pseudonyms.

Throughout the presented work we extract an implicit requirement of the solution, where terminal support must be present for optimal privacy, which is a perfectly valid requirement in the context of SWIFT, but not with the present use of IdM solutions as up until now the driving use case for SAML and OpenID was terminal transparent IdM. However, we still presented solutions that circumvent the terminal support limitation and still provide an acceptable degree of privacy, as shown by the presented use-cases. Another

requirement under evaluation concerns the potential bottleneck introduced by the network stack which may prevent us from having a direct mapping between SAML pseudonyms and virtual network stacks. While we acknowledge its importance and potential downside, we are still conducting the experimentation and theoretical analysis of such impacts.

Through IdM concepts we have aligned pseudonymity at the network stack, with IdM pseudonyms, making the whole greater than the sum of the parts. While this does not come without costs in terms of performance and requirements, we perceive this to be a suitable compromise that harbours considerable gains in terms of user privacy. Furthermore we envision this as being aligned with IdM policies (as is the case of SWIFT), allowing the cost/gain relation to be fine tuned.

Beyond the aforementioned performance and impact evaluation we are considering as part of the ongoing work, how the defined countermeasures for privacy are influenced or defeated by tracking end-user actions and by user (identity) profiling, where several identities could be correlated by performing the same actions (like movement) or by always appearing together in the network. However, such scenarios require elaborate attacker models, given that several entities (on different stratum) must collude to carry out such privacy invasions.

References

- [1] Girao, J. (ed.), "SWIFT, Deliverable 203, First Draft of the Identity-driven Architecture and Identity Framework," 2008.
- [2] S. Cantor, J. Kemp, R. Philpott, and Eve Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0." <http://docs.oasis-open.org/security/saml/v2.0/>, March 2005.
- [3] J. Girao, Alfredo Matos, S. Sargento, and R. L. Aguiar, "Preserving Privacy in Mobile Environments With Virtual Network Stacks," in 50th Annual IEEE Global Telecommunications Conference, (Washington, DC, USA), GLOBECOM 2007, November 2007.
- [4] A. Zugenmaier, "The Freiburg privacy diamond," Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, vol. 3, pp. 1501–1505 vol.3, Dec. 2003.
- [5] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo, "Security assertion markup language (SAML) v2.0 technical overview working draft 21 February 2007." <http://www.oasisopen.org/committees/download.php/22553/sstc-saml-tech-overview-2>
- [6] A. Sarma, A. Matos, J. Girão, and R. L. Aguiar, "Virtual identity framework for telecom infrastructures," Wireless Personal Communications, vol. 45, pp. 521–543, June 2008.
- [7] Marx, R. (ed.), "SWIFT, Deliverable 302, Specification of General Identity-centric Security Model that supports user control of privacy," January 2009.