SWIFT: Advanced identity management

Elena Torroglosa, Alejandro Pérez, Gabriel López, Antonio F. Gómez-Skarmeta and Oscar Cánovas
Department of Information and Communications Engineering
University of Murcia

***Abstract*- Identity Management (IdM) emerges out of the necessity to establish control of digital identities, protecting personal information and establishing confidence in the services providers. The SWIFT framework goes a step further of the traditional IdM solutions, and provides an environment for an advanced management of end users identities. This framework will mainly provide end users identity aggregation from different individual identities, anonymous services access and cross-layer authentication and authorization.**

Keywords. Identity Management, user centric, privacy, anonymity, access control, SSO, cross-layer.

I. INTRODUCTION

The Digital Society is characterized by an intensive use of the Internet where users make constant use of different services in the network, accumulating multiple accounts to be identified in services and social networks. Those service accounts involve a large number of user names and passwords that become totally unmanageable, which also implies the dissemination of personal information, which must be protected.

An individual digital identity collects information that characterizes the individual from the rest. A subset of information can define partial identities on specific context within a domain, as a role for age, nationality or their preferences. Hence, a partial identity can be used for work, another for social relationships, or hobbies.

Identity Management (IdM) [1] emerges out of the necessity to establish control of these digital identities, protecting personal information, establishing confidence in the services (Web and network access) and electronic transactions.

Currently, there are a lot of services that offers private and public solutions that manifest Identity Management capabilities. This fast growth is due to the development of proprietary solutions like OpenId [2] and Microsoft CardSpace [3], and the work done by diverse standardization forums [4].

SWIFT framework [5] goes a step further, and provides advanced identity management solutions like identity aggregation, anonymity, cross-layer SSO, advanced access control and mobility at network level, and between devices.

In order to describe these characteristics, this paper is structured as follows. Section II offers an overview of the SWIFT objectives and how it incorporates this set of advanced identity management solutions. In section III, components of the architecture are briefly described, while section IV depicts some use cases illustrating the basic behaviour of the different entities. Finally, we end with some conclusions.

II. SWIFT OBJECTIVES

The SWIFT project aims of delve into advanced identity management, privacy and anonymity control, and SSO mobility among other topics. The following sections describe the most important lines of research.

A. Identity aggregation

SWIFT provides users with the ability to aggregate different identities into one identity, called virtual identity, which enables a user to group authentication credentials and attributes associated with each of her individual identities. These virtual identities are created in special identity providers called Identity Aggregators, which act as intermediaries between service providers and the real identity providers.

Each user can define as many virtual identities as she wants, adding different attributes and credentials from different providers. Each virtual identity has a virtual identifier (VID) which is unique for each virtual identity at its Identity Aggregator. This VID is used by the end-users to refer to their virtual identity when they interact with the Identity Aggregator and service providers.

When a virtual identity is created, the user links authentication credentials and attributes from different identity providers (Figure 1.). According to the type of information they supply, the providers can play different roles. When providing authentication credentials, they are called Authentication Providers, and when they supply attributes, they play the role of Attribute Providers. These functions are not mutually exclusive and can be performed by a single provider. To be able to create identities as aggregation of other offers more flexibility, simplicity and potential to the end users. The user, as central figure of the system, is responsible for creating and managing her virtual identities.

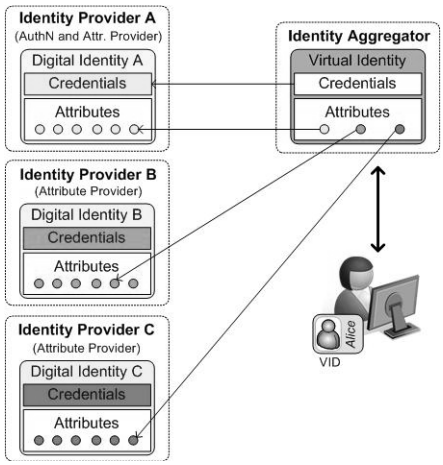

Figure 1. Virtual identity aggregation.

CHINACOM 2010, August 25-27, Beijing, China

DOI 10.4108/chinacom.2010.91

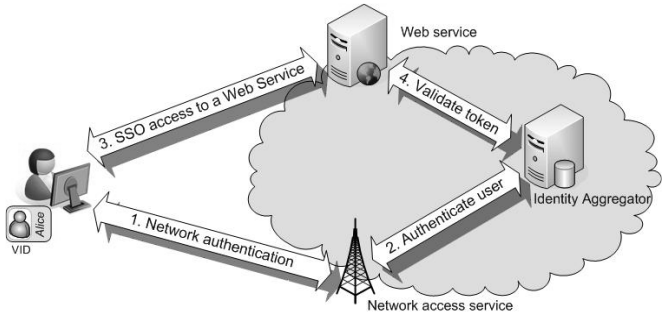

Figure 2. Cross-layer SSO access.

B. *Cross-layer SSO*

One of the main objectives in SWIFT is to integrate network access with high level services. This integration goes from the network level authentication to access these high level services making use of VIDs.

As Figure 2. depicts, SWIFT enables authentication at network layer to also be valid at the level of web services. Until now, these two layers were kept apart, but with SWIFT, not only the SSO [8] is achieved between the upper layer services, but the material from the network authentication allows user to remain authenticated when she subsequently accesses different service providers, offering identity federation.

C. *Privacy management*

One of the main cornerstones of the SWIFT framework is user privacy control. On one hand, we want to protect user identity and traceability, allowing anonymous access to services by means of the VIDs. On the other hand, we want to provide end users with the ability to decide what kind of personal information or attributes will be disclosed to a particular service provider.

In the first case, SWIFT protects end user VID so that only their Identity Aggregator can understand them. Even though that information passes through such entities as service providers, these entities are unable to understand or relate these protected VIDs from other user accesses. The use of pseudonyms, permanent or transient, between service, aggregators and identity providers, help to keep end user anonymity.

On the second case, users should be able to control what information they want to disclose and to whom. When the user accesses different services, the aim is to disclose as little personal information as possible. To that end, SWIFT makes use of advanced attribute release policies. Identity Aggregator is at the core of trust in the SWIFT framework. It is the only entity that can associate the end user's virtual identity with her real identity on an identity provider.

D. *Authorization based on end user attributes*

In the SWIFT framework, service access control is based on the attributes and credentials aggregated by the users to their virtual identities. In first place, the user must be authenticated by her Identity Aggregator (based on a valid authentication process made by user's Authentication Provider). But after the authentication process, service providers may need some user attributes to authorize access to the services.

SWIFT provides a framework for advanced access control [6][7] based on authorization policies [7], which allows service providers to take into account the user attributes (e.g., home organization, age, role,) to make access control decisions. As described in section C, users could set what attributes can be submitted to a particular service provider. Attribute access is made through identity aggregators, allowing service providers to access user attributes without more information about them.

Besides access control, user attributes allow service providers to offer customized services or differentiated QoS, based on the obtained information such as the role of the end user on her company, the type of subscription she has on the provider, her preferences, etc.

E. *Virtual terminal*

Nowadays, end users access to Internet and its services from heterogeneous devices. These devices range from a mobile phone to a laptop. The use of identities should not be restricted to one particular device, which entails that the user should be available to use her virtual identity from any device.

SWIFT is working on the definition of the virtual terminal concept [8] with the aim to abstract the end user virtual identity and the related session from the specific terminal. This technology allows users to discover, share and transfer users sessions between different terminals thanks to the virtual terminal tools. One requirement is that the user establishes a trust relationship between the devices beforehand.

F. *Mobility*

Associated with the virtual terminal concept introduced above, SWIFT covers other aspects related to mobility [9]. SWIFT identity management mechanisms facilitate the mobility management to enable SSO [10] in the network access.

SWIFT framework provides support for user mobility level at network access, taking even into account when the user switches to a non SWIFT-enabled network provider. In this case, SWIFT is able to generate adequate credentials using the credential bootstrapping mechanism [11]. It also takes into account the dynamic information of the user based on their location or connection. This allows, among other things, that service providers can offer more customized services to the user.

III. OVERVIEW OF THE SWIFT ARCHITECTURE

The previous section has described a series of important aspects and objectives within the SWIFT framework. In this section we briefly present an overview of the SWIFT architecture, including the main entities that perform the

described functionality and the different statements that are exchanged among them.

They are:

- **Identity Aggregator (IdAgg):** This is the central entity of the architecture. It is the centre of trust for the components and acts as a mediator between the user, the services and the identity providers. Its main functions are the management of the virtual identity life-cycle, the control over the authentication and SSO mechanisms and the mediation in the attribute retrieving process.
- **Authentication Provider (AuthNP):** It is responsible for the user authentication and for the provision of that information to the IdAgg. It can allow different authentication methods (SIM card, login/password, digital certificates, etc.).
- **Attribute Provider (AttP):** It manages end user identity information in form of attributes. These attributes are stored and distributed to the IdAgg when they are requested.
- **End User (EU):** This is the end user of the system that desires to access to the services offered by the different providers. She has the control over her different identity accounts, including her virtual identities. She is concerned about her privacy and anonymity when accessing services.
- **Service Provider (SP):** Provides services to End Users. In the scope of SWIFT, a service is anything that provides an additional value to the user, including the network access service itself.

This identity management framework defines five different statements that are used in the interactions between the elements defined above, depending on the action that is being performed on each moment. Each statement contains a validity period that allows determining whether the received or stored statement is still usable or not. Besides, a nonce payload is added to the statement in order to assure its uniqueness and to avoid relay attacks, since each statement must only be used once. Following we provide a detailed description of each one of these statements.

- **Initiation Statement.** This statement is generated by the EU to indicate her determination to start a new authentication process when accessing a service offered by a SP. It contains information about the virtual identity to be used for the authentication and authorization processes, and the SP being accessed. Additionally, as described before, a nonce payload is added to provide uniqueness to the statement.
- **Authentication Statement.** This statement serves as a proof that an entity (issuer) gives to another entity (the recipient) about the authentication status of the referred EU, identified by a pseudonym established between them. Therefore, the statement is composed by the identifier of the issuer, the identifier of the recipient, the pseudonym and a nonce payload to assure the uniqueness of the statement.
- **SSO Statement.** This statement is the main piece within SWIFT to provide the SSO mechanisms, since its possession demonstrates that the EU has been authenticated in a previous interaction with the IdAgg and the AuthNP. This statement is composed by the authenticated EU's VID and an evidence of the performed authentication, which is only known by the EU and the IdAgg that generated it.
- **SSO Token.** This statement is generated and used by the EU when accessing a service offered by a SP in order to make use of the SWIFT SSO mechanisms, that is, without involving the execution of a new authentication process. It is very similar to the Initiation Statement presented above, but in this case the statement also contains the evidence that has previously been received in the SSO Statement from the IdAgg. With the inclusion of this evidence the EU is asserting her authenticity. This statement is composed by three main pieces of information: the VID the EU desires to be authenticated with, the identifier of the SP being accessed and the evidence of authentication extracted from the SSO Statement. Additionally, as described above a nonce payload is added to provide uniqueness to the statement.
- **Attribute Statement.** An attribute statement serves as a proof that an entity (issuer) gives to another entity (recipient) about the identity information of the referred EU, identified by a pseudonym. The statement is composed by the identifier of the issuer, the identifier of the recipient, the pseudonym of the EU, a set of attributes representing the identity information being collected and a nonce payload to assure the uniqueness of the statement.

IV. USE CASES

In this section three different use cases are described, one for the web based authentication, one for the SSO authentication and the last one for the attribute retrieval. These use cases show the functionality of the framework, based on the interactions that occur between the entities to perform some of the most usual identity management actions. Additional use cases, including the one for network access authentication, can be found in [5].

A. Web based authentication

This use case describes the situation when an unauthenticated EU (Alice), owning a virtual ID, wants to access a web service provided by a SP. Web redirections are used to carry out the process of authentication. Figure 3. depicts this use case.

Alice wants to access a web service provided by SP. As an initial step, she generates an Initiation Statement including the VID she desires to use for the access and the SP being accessed. Then, she contacts the SP and delivers it a new Access Request message including the newly generated statement along with the identifier of the IdAgg that manages the virtual ID (1). With this information, the SP is able to redirect Alice to the indicated IdAgg, including an

Authentication Request message, which transports the received Initiation Statement from the user (2).

When the IdAgg receives this message, looks for the virtual ID referenced by the received VID and determines the AuthNP and pseudonym that have to be used to authenticate Alice. Then the IdAgg redirects Alice to the AuthNP, including a new Authentication Request message (3) that indicates the pseudonym of the user to be authenticated. The AuthNP prompts Alice for her user name and credentials (4). If the authentication is successful the AuthNP generates an Authentication Statement, where it is asserted that Alice has been correctly authenticated. Then she is redirected back to the IdAgg including this statement along with the redirection. (5).

When the IdAgg receives the statement, it generates a new one for the SP, as well as a SSO Statement for the EU. The new Authentication Statement contains a pseudonym that must be used by the SP to refer to Alice. Then the IdAgg redirects Alice to the SP including these two statements (6), though Alice extracts the SSO Statement before reaching the SP (7).

At this point, the SP is aware that Alice has been successfully authenticated by a trusted AuthNP, but without knowing which provider was and the user name of Alice. Now, the SP can perform an authorization process (8) (as will be described in detail in section IV.C) and provides the service to Alice (9).

This use case accomplishes the identity aggregation (II.A) and privacy management (II.C) objectives, since a virtual identity is used to access the service, while Alice's actual identity is not revealed to the SP.

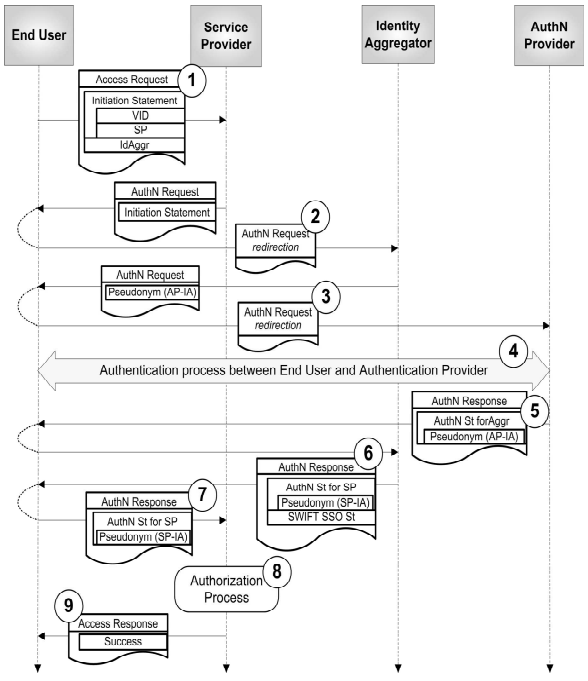

Figure 3. Web authentication use case.

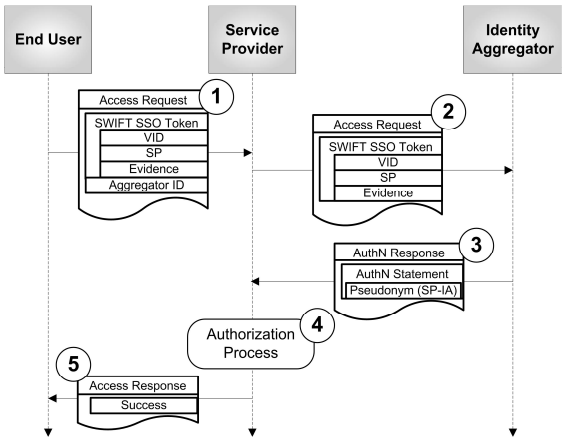

Figure 4. SSO authentication use case.

B. *Single Sign On authentication*

As explained before, this framework provides a SSO mechanism to avoid Alice to be re-authenticated on every access to a service. This SSO mechanism is based on the distribution of the SSO Statement from the IdAgg to the EU along with the first authentication process. With this piece of information, Alice is able to build proper SSO Tokens presented to the different SPs that Alice wants to access. This use case describes the interactions that should be performed among the framework entities to authenticate the EU based on these SSO Tokens. Figure 4. depicts the process of accessing a service by means of the SSO mechanisms.

Alice decides to access a SP (the service can be either a web service or a network service). As she was already authenticated by her IdAgg, she does not want to repeat the authentication process again. Hence, she generates a new SSO Token instead of an Initiation Statement to be presented to the SP. This token contains the same information as the Initiation Statement, but it also includes the Evidence received in a previous step during the first authentication Alice performed with her IdAgg. This SSO Token is sent to the SP along with the identity of the IdAgg where it has to be validated (1). Once received, the SP forwards the SSO Token to the IdAgg in order to verify it (2).

The IdAgg verifies that the Evidence corresponds with the VID and, if so, generates a new Authentication Statement for the SP (3). This statement contains a new pseudonym that must be used between the SP to refer to Alice. When the SP receives this Authentication Statement, it can safely grant the EU to access the service, since the IdAgg has asserted Alice's identity. In a similar way than in the previous use case, the SP can now perform an authorization process (4) and provide the service to Alice (5).

This use case accomplishes the identity aggregation (II.A), cross-layer single sign on (II.B) and privacy management (II.C) objectives, since a virtual identity is used to access the service by means of the SSO mechanism, and still Alice's actual identity is not revealed to the SP. Besides, the SSO mechanism can be used independently of the network layer being accessed.

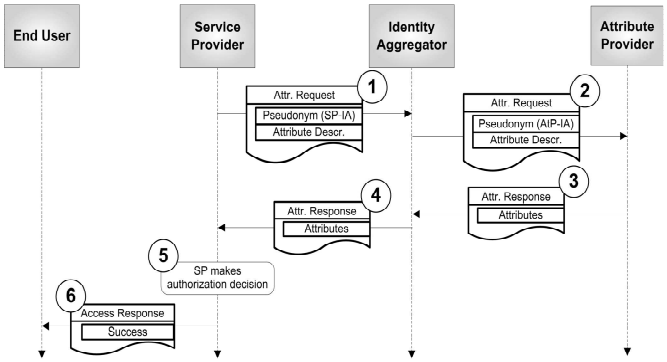

Figure 5. Attribute retrieval and authorization use case.

C. *Attribute retrieval and authorization*

In order to be granted to access a service, Alice may require not only to be properly authenticated, but also to fulfil a series of requirements. This is called the Authorization process. For example, a SP might require Alice to be over a determined age, to be from a specific organization or to have a valid credit card number. That is, the SP can impose certain restrictions on Alice's identity information prior to grant her to access the service. The SP can retrieve this information, in form of attributes, from the IdAgg. These attributes are requested making use of the pseudonym received along the authentication process. Nevertheless, Alice could define access policies to decide which SP is allowed to access to what information. These policies are called Attribute Release Policies (ARP).

This use case, depicted in Figure 5. , describes the interactions that should be performed among the different framework elements to allow the SP to obtain Alice's attributes based on her virtual identity information.

Once Alice has been authenticated, for example making use of one of the authentication processes described above, the SP can retrieve some of her attributes to decide whether she is authorized or not to access the service. In order to do that, the SP constructs an Attribute Request message containing the pseudonym that identifies Alice between the SP and the IdAgg, and the list of required attributes (1). When the IdAgg receives such a request, it looks for the virtual identity referenced by the indicated pseudonym and checks whether the requested attributes were aggregated by Alice into that identity. If so, the IdAgg verifies, based on the ARPs previously defined by Alice, whether the SP can access the requested attributes or not.

Thus, if the SP is allowed to access them, the IdAgg contacts with the different AttPs that actually store the attributes, in order to retrieve them (2). To do that, the IdAgg makes use of the pseudonym established between them during the enrolment process to reference Alice. Each AttP provides the requested attributes to the IdAgg included into an Attribute Statement (3). Once the IdAgg has collected all the attributes requested by the SP, it generates a new Attribute Statement containing all the information and delivers it to the SP (4).

When the SP receives all the attributes, it can perform the authorization process and determine whether the EU can access the service or not (5). In order to perform this authorization process, the SP is subdivided following the architecture defined for XACML [7], including its different entities (PEP, PDP, etc.) and the communication among them. These interactions are not described here since it is out of the scope of this document. Finally, the SP provides the service to the EU (6).

This use case accomplishes the identity aggregation (II.A), privacy management (II.C) and authorization based on attributes (II.D) objectives, since a virtual identity is used to obtain the required attributes to perform the authorization process, while Alice's actual identity is not revealed to the SP.

V. CONCLUSIONS

Service administrators have to be very careful to provide end user new attractive services, but taking special attention to protect the end users identity and their related information. SWIFT provides a framework able to deal with advanced identity management services ranging from an access control mechanism able to take into account the end user attributes, SSO authentication through different network layers, to anonymous access and privacy. Besides, the framework provides end users identity aggregation from different individual identities, anonymous services access and cross-layer authentication and authorization.

The current project work includes among other activities: a formal specification of the core framework, a formal security analysis with AVISPA tool, and the design and development of several prototypes for the different use cases.

ACKNOWLEDGMENTS

This work was partially funded by SWIFT (FP7, Grant Number 215832). The authors also thank the Funding Program for Research Groups of Excellence (04552/GERM/06) established by Fundación Séneca.

REFERENCES

[1] Maler, E. and Reed, D. 2008. *The Venn of Identity: Options and Issues in Federated Identity Management.* IEEE Security and Privacy 6, 2 (Mar. 2008), 16-23.

[2] OpenId, http://openid.net/

[3] Microsoft CardSpace. Webpage: http://msdn.microsoft.com/en-us/windows/aa663320.aspx

[4] Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org, 2009

[5] Gabriel López, Cánovas, Antonio F. Gómez Skarmeta, Joao Girao, *"A SWIFT Take on Identity Management*", IEEE Computer, Vol 42 (May 2009), 58-65

[6] OASIS, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* – Working Draft, February 2007

[7] OASIS. *eXtensible Access Control Markup Language (XACML)* Version 2.0, 2005.

[8] SWIFT, D203 *"First Draft of the Identity-driver Architecture and Identity Framework"*

[9] SWIFT, D402 *"SWIFT Mobility Architecture"*

[10] M. Sánchez, G. López, O. Cánovas, and A.F. Gómez-Skarmeta. *Bootstrapping a global SSO from network access control mechanisms.* In Proceedings of the 4th European PKI Workshop (EuroPKI'07, June 2007. Lecture Notes in Computer Science (LNCS), Volume 4582/2007.

[11] SWIFT, *White Paper "Identity as a Convergence Layer"* https://www.ist-swift.org/component/option,com_docman/task,cat_view/gid,34/Itemid,37