Efficient Application SSO for Evolved Mobile Networks

Andreas U. Schmidt, *Member, IEEE*, Andreas Leicher, Novalyst IT AG
Yogendra Shah, Inhyok Cha, Interdigital Communications Inc.

***Abstract*—Efficient and seamless Single-Sign-On (SSO) access to applications is a core question for the evolution of services in wireless networks. A fundamental requirement is lightweight, secure authentication protocols and an overarching identity management framework. We review 3GPP standardisation efforts on this topic, and propose generation of identity assertions, locally at the device as an efficient and secure authentication method for mobile SSO.**

***Index Terms*—Single-Sign-On, OpenID, GBA, 3GPP.**

I. Introduction

Single-Sign-On (SSO) is the strongest, and most user-centric, form of combined authentication and authorisation (AA). It assumes that a user has to log on only once to an *SSO domain*, providing his/her credentials such as user name and password, and then has continuous access to all *affiliated services (AS)* of that domain. The log on remains valid until a designated log off event occurs or, for instance, the registration expires after a time limit is reached. A genuine architecture for SSO systems is centered on a *domain controller* which facilitates AA toward services via *delegated authentication*. That is, upon a service access attempt, an AS requests confirmation of the log on status and authentication of the requesting user/client from the domain controller. The role of a domain controller in SSO is active since he is involved in the actual authentication protocol. This is in contrast to the passive role of trusted third parties (TTPs) in PKI, for example, where certificates speak for the provider of an identity, e.g., a certification authority.

SSO is ubiquitous and known to everyone using a Windows Server domain. The domain controller and client log on application, called LSASS, run a challenge-response protocol called NTLM (v2) on every service request, in which LSASS uses hashed passwords as credentials. NTLM v2 also allows integrated use of the ticket system Kerberos [1]. In communication networks, SIP registration with an IMS domain [2] is one realisation of SSO.

Due to the seamless bundling of authentication toward multiple AS, SSO raises some critical security concerns. The AS of a domain often share no existing trust relationship, and may vary in requirements on authentication strength. Furthermore, proliferation of credentials and breaking of device binding are concerns. In [3,4], we have considered Trusted Computing as a means to endow User Equipment (UE) with trustworthiness in Kerberos, and OpenID based authentication, respectively. These lightweight delegated authentication protocols, in particular OpenID, are promising candidate technologies for the authentication part of SSO, if they are combined with a domain control framework.

In mobile networks, control over a subscriber domain is naturally exerted by the Mobile Network Operator (MNO), and it may rely on a hardware security anchor of a subscription, embodied by the UICC. The present contribution reviews the standardisation approach and activities of the 3rd Generation Partnership project (3GPP) to application SSO, highlighting the particular requirements of mobile networks. The first main question to answer is the secure linking of subscriber authentication to the SSO system, for instance in a combined Generic Bootstrapping Architecture (GBA, cf. [5]) – OpenID protocol. The second one is an overarching framework to join SSO domains, and thus underlying subscriber identities, for instance IMS and non-IMS domains. Finally we present the concept of a *local assertion provider* as a particularly efficient protocol for SSO in wireless networks, based on OpenID, but applicable to general network-centric SSO.

II. 3GPP Standardisation

The security working group SA3 of 3GPP has made several approaches to incorporate user sign-on and identity management (IdM) technologies in their portfolio of specifications. Attacking the problem from the viewpoint of MNOs, SA3's strategy is to bind existing Web IdM solutions such as Liberty Alliance (LAP) and OpenID [6] to 3/4G network authentication, in particular GBA.

In a technical recommendation drafted between 2005 and 2007 [7], SA3 describes the interworking of GBA with LAP Identity Web Services Framework. The core idea is very simple: Credentials for use in the LAP IdM framework are bootstrapped from the GBA functions of the network, i.e., between the network elements Network Address Function (NAF), Bootstrapping Server Function (BSF), Home Subscriber Server (HSS) and the UE with the UICC on the other side. To maintain trust, the identity provider of the LAP IdM framework is co-located with the NAF or BSF.

The above paradigm is extended in Technical Recommendation 33.924 [8] to OpenID Web authentication. OpenID is attractive due to its steep uptake by major Web services such as Google and Facebook, its distributed architecture, which is appealing to MNOs, and its simple deployment. Again, the natural approach taken in [8] is to leverage mobile network authentication by integrating the OpenID identity provider with the NAF. We provide more details on OpenID and OpenID/GBA in Section II below.

The commonality of LAP/GBA and OpenID/GBA is the UICC. Acknowledging that not all authentications in evolved networks may rely on UICCs, and that SSO should optimally function across a diversity of underlying authentication methods, SA3 recently commenced a new study item. It explicitly targets [9] "to describe the re-usage of non-UICC credentials, in particular SIP Digest credentials, to provide security for access to applications." Such credentials are in particular used in the context of registration of UE in IMS (Internet Multimedia Subsystem) networks [1, Annex N].

The study takes a more generic approach toward the coupled SSO or IdM technology than the point solutions LAP/GBA and OpenID/GBA (see Figure 1). It envisages a central role for the network's HSS as a connector between the SSO domains, via a to-be-specified SSO-specific interface. In this way, a log-on at the IMS domain may be automatically carried forward to any connected SSO domain, for instance to Web Services using SSO and accepting authentication and/or UE enrolment via the HSS interface.

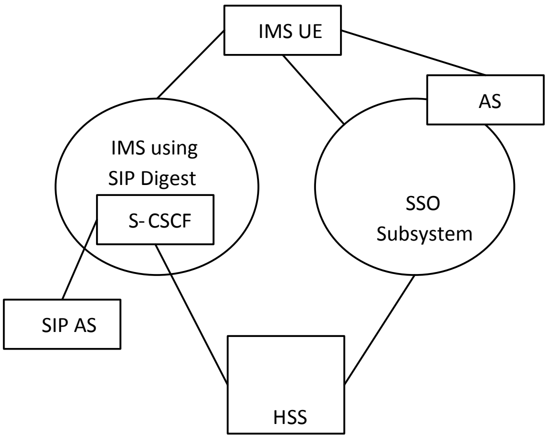

Fig. 1. IMS-SSO Interworking High-level Architecture [9].

Though more generic than the previous solutions, the IMS-SSO study [9] is still limited in scope, this time to UICC-less authentication. The diverging directions of work in SA3 lately prompted the services and features working group SA1 to adopt another fundamental work item on SSO [10], tasked to enable the "Integration of Single Sign-On (SSO) to 3GPP Services." The objectives are:

- Comprehensive set of use cases covering the different Identity and SSO frameworks with focus on OpenID
- Analysis of use cases for SSO/OpenID integration with the operator core network with or without GBA
- Study requirements for leveraging operator controlled user credentials and trust framework with 3rd party service providers
- Focus on SSO integration for different access technologies and identity credentials
- Service level interworking and UE requirements for these use cases
- Consideration of both IMS (including common IMS) and non-IMS service architectures
- Consideration of existing authentication and bootstrapping mechanisms (e.g. AKA, GBA, SIP-Digest, etc).

Some salient requirements for network-based SSO systems have been carved out in the work of 3GPP sketched above. Apart from architectural requirements, for instance interworking with a large range of SSO systems and authentication methods, and interfacing with non-3GPP standard protocols, efficiency and security requirements are essential guidelines for a successful SSO standard. In particular, the core network elements involved in the SSO procedures should bear only the minimum necessary load, and they should protect the secrets with which they operate in these processes. Also maintaining the separation of identity domains across which the operator authentication federates (Fig. 1), is important for user privacy. Some design requirements [11] for IMS-based SSO capture these aspects:

- Where user privacy is required, the design of a SIP Digest based SSO system should not allow affiliated non-IMS domain services to draw conclusions about IMS domain identities, e.g., the SSO subsystem should hide IMPIs [IP Multimedia Private Identities] from application services.
- Any solution should take into account the following design guidelines for HSS-related security:
  - The number of different types of interfaces to the HSS should be minimised in order to keep the complexity of the HSS low. This applies in particular to interfaces over which authentication vectors are retrieved from the HSS as they are highly security-critical.
  - In order to minimize any security risks due to excessive use/abuse of authentication vectors, as well as any performance impact to HSS and AuC [Authentication Centre], the overall number of authentication vectors requested from the authentication centre as well as the number of requests should be kept low. Mechanisms which make economical use of authentication vectors should be preferred. In particular, mechanisms which avoid bursts in authentication vector requests should be preferred.

This hints to the complexity of integrating IdM and SSO into 3G network architectures which are not per se designed for such tasks. In the following we consider OpenID/GBA in more detail and propose one solution that help satisfy the afore-mentioned requirements.

III. OpenID and OpenID/GBA

This section gives a brief overview over the OpenID protocol and its coupling to GBA-based network authentication as specified by SA3.

A. *OpenID Fundamentals*

As an open, decentralized IdM framework, OpenID [6] was developed to provide a SSO experience to users across services on the Internet. With OpenID (see Figure 2), it is possible to sign on to different services, with a single identity, called OpenID identifier, eliminating the need to create separate login and password credentials for the services the user wants to access. OpenID is supported by major companies, including AOL, Facebook, Google, Microsoft, Yahoo, etc. Reports of OpenID usage [12] count over 1 billion OpenID enabled accounts and over 9 million websites utilizing OpenID for registration and login. Recently, efforts were made by the OpenID Foundation and the US government [13] to deploy OpenID on federal websites.

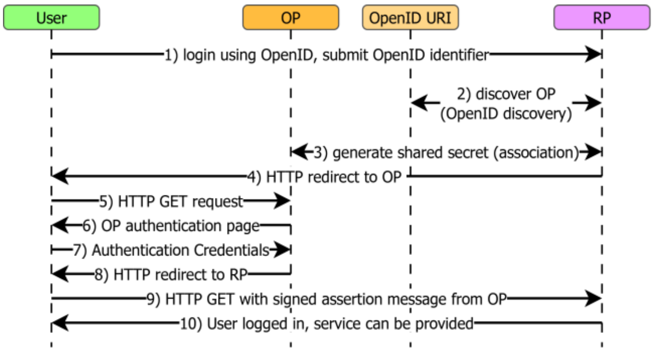
User
OP
OpenID URI
RP
1) login using OpenID, submit OpenID identifier
2) discover OP (OpenID discovery)
3) generate shared secret (association)
4) HTTP redirect to OP
5) HTTP GET request
6) OP authentication page
7) Authentication Credentials
8) HTTP redirect to RP
9) HTTP GET with signed assertion message from OP
10) User logged in, service can be provided

Fig. 2. OpenID Protocol Overview.

The websites supporting OpenID login are referred to as Relying Parties (RP), the TTP to which authentication is delegated is the OpenID Provider (OP). OpenID solely relies on the HTTP protocol for message transport. For this reason, user identifiers in OpenID are represented by URLs. A slightly simplified OpenID protocol run (in the so-called association mode) is shown in Fig. 2.

B. *OpenID/GBA TR 33.924*

In OpenID/GBA, the network authentication infrastructure is used directly for each log-on using OpenID toward an RP. For this, the OP is co-located with the NAF as shown in Figure 3.

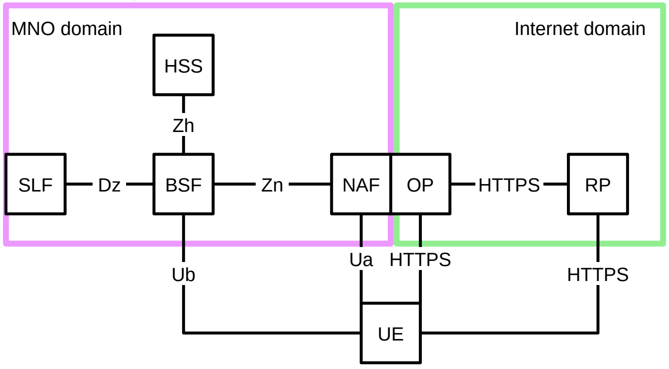

Fig. 3. OpenID/GBA Architecture According to [8]
(For the GBA notions on the left hand side see [5])

The authentication of the UE (e.g. UICC) via GBA is done inside the OpenID protocol run (between steps 5 to 8 in Figure 2). Although simple and elegant, OpenID/GBA has some drawbacks. Most importantly, each OpenID authentication request triggers a GBA run, which puts a burden on the network infrastructure, contrary to the requirements mentioned in the last section. This may also have security implications. An attacker may submit indiscriminate HTTP GET requests (for instance using a spoofed IP address of a target UE) to trigger frequent GBA runs between that UE and the NAF/BSF. This may constitute a denial-of-service attack on UEs, and, on a larger scale, the GBA infrastructure. In general, the traffic pattern from OpenID/GBA runs (Web authentication) may be very different from the authentication traffic for UE network attachment.

IV. Local Assertion Providers

This section describes our proposal of a *local assertion provider* for network-centric SSO. We first present the general rationale and idea and then detail a protocol adjustment of OpenID/GBA that includes the local assertion provider. Finally, security, efficiency and implementation/deployment options are discussed.

A. *Basic Concept*

Previous proposals [7, 8] for network-centric SSO follow a paradigm of direct coupling of delegated authentication to the networks AA (Authentication and Authorisation) infrastructure, entailing the problems described in the last subsection.

In hindsight, this solution seems sub-optimal leading us to the idea of shifting the authentication paradigm from “tight coupling” to “loose coupling.” Loose coupling is a term describing the protocol and interface relationships between entities in Service-Oriented Architectures (SOA). It refers to the degree of knowledge (data) that one entity has to have about another one, to communicate correctly with it [14]. For instance, OpenID/GBA represents tight coupling between MNO domain entities and Internet domain entities (Figure 3), since the former domain requires full information about each OpenID run in the latter domain to authenticate a UE. On the other hand, OpenID itself is a loosely coupled Web protocol between RP and OP via the discovery process (step 2 in Figure 2) – any RP does not need to know (or learn from the client) fixed OP addresses to initiate the authentication. Loose coupling has also been used as a paradigm for handover between heterogeneous access networks in 3G systems [15].

To implement loose coupling in SSO using a TTP for delegated authentication – such as the OP/NAF of OpenID/GBA – authentication of UE by that TTP needs to be decoupled from the provisioning of confirmations about successful authentication to the AS – e.g. the sending of signed OP assertions to the RP. To enable this, we introduce the concept of a *local assertion provider* in the UE. The local assertion provider has authority to sign assertions *in lieu of* the TTP. The local assertion provider gains authority to sign assertions from the SSO schemes’ root TTP in a provisioning process, in which the UE performs a proper authentication toward the TTP, and the local assertion provider is thereafter, endowed with secrets to sign subsequent identity assertions (delegation of authority). Although the local assertion provider could be localized anywhere, co-locating it with the UE helps to a) de-centralise authentication traffic, and b) maintain authentication strength. In particular, we propose to co-locate the assertion provider in the secure environment of the UICC.

B. *Improved OpenID/GBA Protocol*

The following outlines the local identity assertion methods for OpenID/GBA. In this scenario, a multitude of OpenID authentications to an RP is divided into the phases of provisioning and authentication, where the provisioning phase includes the normal process of OpenID/GBA [8].

1) *Provisioning Phase*

This phase is outlined in Figure 4. It is very similar to the direct OpenID/GBA run of [8, Section 4.1.1] with some additions, marked in italics in the following protocol description and by red ellipses in Figure 4. The original text of [8] is somewhat abridged. This message flow is only executed when the user logs on to an RP for the first time.

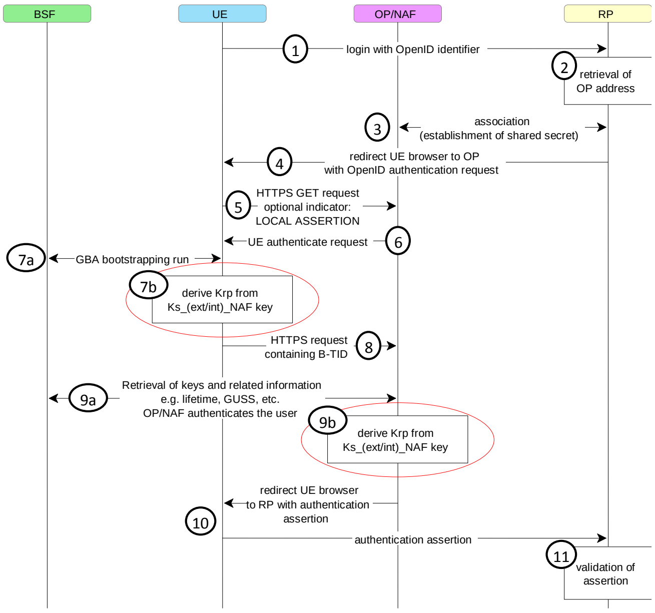
BSF
UE
OP/NAF
RP
1
login with OpenID identifier
2
retrieval of OP address
3
association (establishment of shared secret)
4
redirect UE browser to OP with OpenID authentication request
5
HTTPS GET request optional indicator: LOCAL ASSERTION
6
UE authenticate request
7a
GBA bootstrapping run
7b
derive Krp from Ks_(ext/int)_NAF key
HTTPS request containing B-TID
8
9a
Retrieval of keys and related information e.g. lifetime, GUSS, etc. OP/NAF authenticates the user
9b
derive Krp from Ks_(ext/int)_NAF key
redirect UE browser to RP with authentication assertion
10
authentication assertion
11
validation of assertion

Fig. 4. Provisioning of a Shared Secret to a Local Assertion Provider Inside an OpenID/GBA Run.

1. The Browser Agent (BA) in the UE sends a user-supplied identifier to the RP.
2. The RP performs a discovery of the OP based on the User-Supplied Identifier [6].
3. The RP and the OP may then establish a shared secret (called association). The purpose of this shared secret is that the OP can secure subsequent messages and the RP can verify those messages.
4. The RP redirects BA to the OP with an OpenID Authentication Request defined in [6, Chapter 9].
5. Following this redirection the BA sends a HTTP GET request to the OP/NAF. *In order to* indicate *to the OP/NAF that local identity assertion is supported, the request may contain the indication "LOCAL ASSERTION"*.

Note that this kind of indication may be omitted if the OP/NAF decides on use of local identity assertion based on, e.g., the user supplied identifier (UEs known to have a local assertion provider).

6. The NAF initiates the UE authentication and responds with a HTTPS response code 401 “Unauthorized”, which contains a WWW Authenticate header carrying a challenge requesting the UE to use GBA Digest Authentication.
7. There are two subsequent key derivations to provide shared secrets, first to UE.
   a. If no valid Ks is available, then the UE bootstraps with the BSF [5], resulting in the possession of the UE of a valid Ks. From this, UE derives the application (OpenID) specific Ks_(ext/int)_NAF key(s).
   b. *If no valid RP specific key Krp is available, UE derives Krp from the OpenID specific Ks_(ext/int)_NAF key, resulting in possession of a valid RP-specific key Krp by the UE.*
8. The UE generates a HTTP GET request to the NAF. The HTTP request carries an authorization header containing the B-TID (the unique GBA bootstrapping identifier) received from the BSF.
9. OP/NAF has to undergo the same key derivation as UE to establish shared secrets.
   a. Using the B-TID and NAF_ID the NAF retrieves the shared application specific NAF key and optionally the USS (if GBA_U are used) from the BSF. The NAF stores B-TID, keys, and user supplied identifier to match the OpenID user session and the GBA session.
   b. If no valid RP specific key Krp is available, the NAF derives Krp from the OpenID specific Ks_(ext/int)_NAF key, which results in possession of the same valid RP-specific key Krp by the NAF, as by the UE after step 7b.
10. NAF/OP authenticates the user for OpenID. The NAF redirects the browser to the return OpenID address, i.e., the OP redirects the BA back to the RP with either an assertion that authentication is approved or a message that authentication failed. The response header contains a number of fields defining the authentication assertion which may be protected by the shared secret between OP and RP. The protection is especially important if the OP and the RP do not reside both in the MNO network.
11. The RP validates the assertion, i.e., checks if the authentication was approved. The authenticated identity of the user is provided in the response message towards the RP. If an association was established in step 3, then it is used to verify the message from the OP. If validation of the assertion and verification of the message are successful, then the user is logged in to the service of the RP.

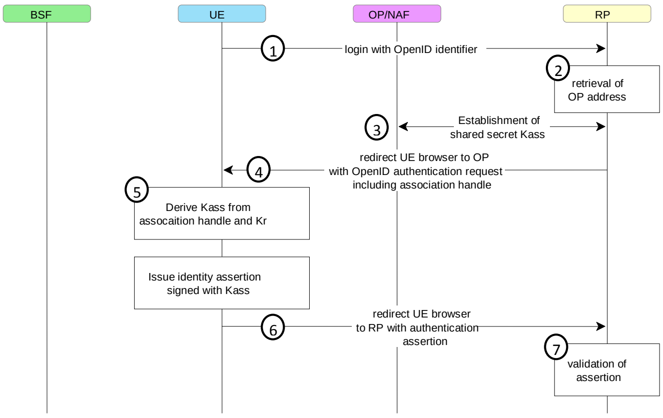
BSF
UE
OP/NAF
RP
1
login with OpenID identifier
2
retrieval of OP address
3
Establishment of shared secret Kass
4
redirect UE browser to OP with OpenID authentication request including association handle
5
Derive Kass from assocaition handle and Kr
Issue identity assertion signed with Kass
6
redirect UE browser to RP with authentication assertion
7
validation of assertion

Fig. 5. OpenID Authentication using a Local Assertion Provider.

2) *Authentication Phase*

Subsequent authentications are now supported by the local assertion provider acting as a proxy for the OP/NAF, this shortens the authentication protocol considerably as shown in Figure 5. This message flow is only executed when the user logs on to a RP after he has logged on to this RP for the first time, which has resulted in sharing of a valid RP specific key Krp, by the UE and the NAF. Steps 1 and 2 are as in the provisioning phase. The remaining steps are:

3. *The RP and the OP/NAF establish an association as [6, Chapter 8]. The association is identified by a unique association handle and shared key generated by the OP. The OP derives a shared key Kass from Krp and the association handle. Kass must follow the specification for OpenID MAC signature keys, i.e. must be a valid key for HMAC-SHA1 or HMAC-SHA256 [8]. The key Kass and the association handle are communicated to RP.*
4. The RP redirects BA to the OP with an OpenID Authentication Request defined in [6, Chapter 9].
5. *The UE derives, from Krp and the association handle contained in the authentication request, the same association specific key Kass as in step 3*
6. *The UE redirects the browser to the return OpenID address at the RP with an assertion that authentication is approved. The response header contains a number of fields defining the authentication assertion which is signed with the shared secret Kass.*
7. The RP validates the assertion as in step 11 of the provisioning phase, *but using the association specific key Kass*.

V. Discussion

Here we discuss the salient traits of the local assertion provider as a basis for network-centric SSO.

A. Security

Due to its inherent properties (de-centralisation), local assertion providers significantly reduce the risks from denial-of-service attacks described in Section III.B. On the other hand, authorising a local entity on the UE to sign identity assertions also has security implications. Namely, the authentication strength in comparison to the original GBA authentication must be maintained in provisioning and authentication phases. Concretely, key derivation, keys and their usage need protection.

The derived keys Krp and Kass are security-critical, as well as the key derivation process. The latter means that knowledge of Kass must not yield information about Krp, and, in turn, knowledge of Krp must not yield information about Ks – a standard security requirement on key derivation, stating the unconditional security of derived keys. This requirement applies to the key derivation of Krp, and subsequently for Kass, at both ends, the UE and OP/NAF. These keys must furthermore be kept in a Secure Storage and Execution Environment (SSEE). As the GBA bootstrapped keys are kept inside the UICC, this solution also suggests itself for Krp and Kass. It should be noted, however, that the risk from leaking Kass is lower than for Krp, since Kass is actually used for only one authentication.

B. Efficiency and Benefits

As shown in the example of OpenID/GBA, local assertion provisioning may significantly improve the efficiency of network-centric SSO. It takes a load off the core network infrastructure which is genuinely not designed to support the expected authentication traffic entailed in Web SSO, and de-centralises the traffic. In fact, the authentication phase protocol eliminates over-the-air authentication traffic to the network AA infrastructure, and reduces the total traffic to a single call over an IP interface (step 3). Therefore, the local assertion provider based protocol better satisfies the requirements set out by 3GPP and mentioned in Section II. The MNO still provides the trust foundation for RPs and controls the user base and may provide branding and seamless SSO as benefits to affiliated Web services, *without incurring significant deployment cost for new infrastructure elements.*

C. Implementation Options

As mentioned, a preferred option for an implementation of a local assertion provider is on the UICC. Considering OpenID/GBA in particular, some implementation requirements emerge. The local assertion provider, in order to be able to act as a proxy for the OP, must speak HTTP(S), and would in the simplest case be reachable from the Internet. A good candidate platform is the Smart Card Web Server (SCWS) specified by the Open Mobile Alliance (OMA). The SCWS is also attractive to integrate user authentication (which is not considered in the OpenID/GBA protocols), e.g., by displaying to the user a trustworthy, well-known login screen, and avoids the sending of user credentials over the Internet (local user authentication). In particular, the SCWS offers the additional security benefits of the UICC in keeping the assertion generation capability within the secure environment of the UICC.

We have performed a local assertion provider feasibility study according to these specifications [16-20], which shows the viability of the concept and implementation on a SCWS.

Acknowledgment

This work was funded by InterDigital, Inc. Special thanks go to Lawrence Case, Bob DiFazio, Dorothy Gellert, Louis Guccione, Dolores Howry, Michael V. Meyerstein, and Christopher Wallace for many useful discussions and comments.

References

[1] N.N., “Kerberos: The Network Authentication Protocol,” http://web.mit.edu/Kerberos/.

[2] 3GPP, “3G security; Access security for IP-based services (Release 10).” 3GPP TS 33.203 V10.0.0, June 2010.

[3] Andreas U. Schmidt, Andreas Leicher, and Nicolai Kuntze, “Implementation of a Trusted Ticket System,” in: *Proc. IFIP Sec 2009 Conf.*, Springer-Verlag, Boston, 2009, pp. 152-163.

[4] Andreas U. Schmidt, Andreas Leicher, “Trusted Computing Enhanced OpenID,“ in: *Proc. ICITST 2010 Conf.,* forthcoming.

[5] 3GPP, “Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 9).” 3GPP TS 33.220 V9.3.0, June 2010.

[6] OpenID.net, “OpenID Specifications.” Available: http://openid.net/developers/specs/. Access date: 18 August 2010.

[7] 3GPP, “Liberty Alliance and 3GPP security interworking; Interworking of Liberty Alliance Identity Federation Framework (ID-FF), Identity Web Services Framework (ID-WSF) and Generic Authentication Architecture (GAA) (Release 9).” 3GPP TR 33.980 V9.0.0, December 2009.

[8] 3GPP, “Identity management and 3GPP security interworking; Identity management and Generic Authentication Architecture (GAA) interworking (Release 9).” 3GPP TR 33.924 v9.2.0, June 2010.

[9] 3GPP, “Single Sign On Application Security for Common IMS – based on SIP Digest (Release 10).” 3GPP S3-100784, June-July 2010.

[10] 3GPP, “WID proposal for Integration of Single Sign-On (SSO) with 3GPP services.” 3GPP S1-102186, August, 2010.

[11] 3GPP, “Comment on S3-100785 Requirements for Application SSO.” 3GPP S3-100887, June/July 2010.

[12] OpenID.net, “OpenID - Year 2009 in Review.” Available: http://openid.net/2009/12/16/openid-2009-year-in-review//. Access date: 18 August 2010.

[13] OpenID Foundation (OIDF), “Open Trust Frameworks for Open Government: Enabling Citizen Involvement through Open Identity Technologies.” Available:http://openid.net/docs/Open_Trust_Frameworks_for_Govts .pdf. /. Access date: 18 August 2010.

[14] Cesare Pautasso, Erik Wilde, “Why is the web loosely coupled?: a multi-faceted metric for service design,” in: *Proc. 18th international conference on World wide web*, 2009, pp. 911-920.

[15] H. Wang, A. R. Prasad, “Security context transfer in vertical handover,” in: *Proc. 14th Personal, Indoor and Mobile Radio Communications Conf., PIMRC*, IEEE, 2003, pp. 2775-2779.

[16] ETSI , “Smart Cards; Smart Card Platform requirements (Release 9),” ETSI TS 102 412 V9.0.1, January 2010.

[17] OMA, "Smartcard-Web-Server; Approved Version 1.1 – 12 May 2009", OMA-TS-Smartcard_Web_Server-V1_1-20090512-A, 2009.

[18] GlobalPlatform, “GlobalPlatform Card Specification Version 2.2,” March 2006.

[19] ETSI, “Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 8),” ETSI TS 102 221 V8.2.0, June 2009.

[20] ETSI, “Smart Cards; Secure Channel between a UICC and an end-point Terminal (Release 7),” ETSI TS 102 484 V7.0.0, December 2007.