Smart OpenID: A Smart Card Based OpenID Protocol

Andreas Leicher, Andreas Schmidt, Yogendra Shah

Smart OpenID
A Smart Card based OpenID Protocol

Andreas Leicher[1], Dr. Andreas U. Schmidt[1], and Dr. Yogendra Shah[2]

[1] Novalyst IT AG,
Robert-Bosch-Str. 38, 61184 Karben, Germany
{andreas.leicher,andreas.schmidt}@novalyst.de
http://www.novalyst.de
[2] InterDigital Communications, LLC.,
781 Third Avenue, King of Prussia, Pennsylvania, 19406 USA
yogendra.shah@interdigital.com
http://www.interdigital.com

Abstract. OpenID is a lightweight, easy to implement and deploy approach to Single Sign-On (SSO) and Identity Management (IdM), and has great potential for large scale user adoption especially for mobile applications. At the same time, Mobile Network Operators are increasingly interested in leveraging their existing infrastructure and assets for SSO and IdM. In this paper, we present the concept of Smart OpenID, an enhancement to OpenID which moves part of the OpenID authentication server functionality to the smart card of the user's device. This seamless, OpenID-conformant protocol allows for scaling security properties, and generally improves the security of OpenID by avoiding the need to send user credentials over the Internet and thus avoid phishing attacks. We also describe our implementation of the Smart OpenID protocol based on an Android phone, which interacts with OpenID-enabled web services.

Keywords: OpenID, Identity Management, Single Sign-On, Authentication, Smart Cards, GBA

1 Introduction

One of the challenges in the growing use of online services is the management of digital user identities [1]. The issues arising from poorly implemented identity management (IdM) include identity theft, phishing, fraud and lack of privacy. Most services implement proprietary IdM systems, where a password based mechanism is the most widely used method for user authentication. Different solutions, such as Liberty Alliance [2], CardSpace [3, 4], Higgins [5] have been proposed to address the issues arising from the extensive use of username/password authentication. However, most of these protocols and architectures have not seen a high adoption rate by end users and services in the consumer market.

In addition, users are becoming more concerned about their privacy and are less willing to provide personal identity information. This results in the requirement that while providing a convenient and comfortable access to services, the

IdM system has to provide security and privacy at the same time. The current distribution of user personal data, leads to different problems such as inconsistency of data, loss of control as well as multiple authentication and sign-on methods for the services the user wants to access. In our contribution we discuss a novel approach which combines the security of smart cards, the authentication systems of mobile network operators and an open IdM protocol, namely OpenID, in order to address these issues.

1.1 Role of Mobile Network Operators

With the evolution of new technology, new market entrants and changed customer expectations, Mobile Network Operators (MNOs) have to face a change in their traditional business model [6]. The potential for MNOs to exploit and leverage the large amount of user data they possess has been studied as part of the EU FP7 project PrimeLife [7]. The *IdM Enabler* concept has been developed to allow an economic valuation of IdM business models.

Building on their customer relationships and an existing infrastructure for authentication and communication, MNOs are a prime candidate to monetize this data by providing IdM services to third-party service providers [8, 9]. They are in possession of *IdM data assets* , i.e., they can provide user attributes such as address, name, etc. and at the same time they have deployed the necessary *IdM Functional Capabilities* , i.e., the technical capabilities to provide authentication, authorization, accounting, attribute management and policy functions.

1.2 OpenID

OpenID [10] was developed with the goal to provide a lightweight, Single Sign-On experience to users across a wide variety of online services. The OpenID protocol allows users to sign on to different services with a single identifier, where the authentication itself is performed by the OpenID provider (OP). OpenID hence eliminates the need to create separate user accounts for the services, and thus tackles the main issues with classical password based authentications, namely phishing, the reuse of passwords on multiple service sites or the use of "Post-Its" to remember passwords [11, 12].

OpenID uses identifiers in the form of an URL provided by an OP. The OP authenticates the user and validates the user's credentials on behalf of the services, which are referred to as Relying Parties (RP).

To sign in to a RP, the user provides his OpenID identifier to the RP, from which the RP can extract the necessary information about the location of the OP. The OP and RP then establish an association, , i.e., they exchange a shared secret which is used to sign any future communication between the OP and the RP which can be identified by an association handle. Then, the RP redirects the user's browser to the OP login page. This redirection message, called OpenID authentication request [10], includes the user identifier and the association handle. The user authenticates at the OP, using the method provided by the OP, which most commonly is via username and password.

However, OpenID itself does not specify the authentication method, which has led to the integration of different authentication mechanisms, such as smart card based SSL certificates [13], TPM based authentication [14], or 3G network authentication mechanisms such as EAP-SIM [15] or GBA [16].

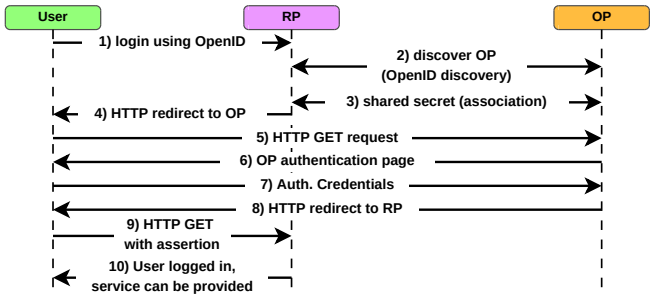

Fig. 1. Overview of the OpenID Protocol Flow

After successful user authentication, the OP redirects the user's browser back to the RP, including a signed assertion on whether the user authentication succeeded or not. The RP can then validate the authentication response by checking the signature on the signed assertion using the association secret. If the check is successful, the service can be provided to the user.

1.3 Smart Cards

Smart cards are portable and tamper-resistant computing devices which are able to securely store and process information. An attacker would need to be in possession of the smart card and also would need additional knowledge of the smart card hardware and software to probe for information. Smart cards are hence often used for secure data storage and authentication. The biggest market for smart cards is the mobile communication industry. Mobile phones have a Universal Integrated Circuit Card (UICC), which is a smart card that identifies the user. The UICC is able to perform authentication algorithms and provides cryptographic functions for encryption and decryption [17]. Using technology like Java Card, it is possible to write Java based code which is then loaded in the form of applets onto the smart card. Communication between a host application and an applet on the smart card uses Application Protocol Data Units (APDUs) in a command/response protocol [18]. Additional APIs, e.g. OpenMobileAPI [19], enable applications running on a mobile phone to access UICC functions.

2 Related Work

As OpenID receives a lot of attention from industry and research, different aspects of OpenID have been discussed in the literature. Most research work is

centered around the integration of different authentication mechanisms into the OpenID protocol. The intent of this section is to discuss the work which is closest to our proposal and highlight how we can improve on existing solutions. Additionally there has been research on weaknesses of the OpenID protocol, which we aim to mitigate by our solution.

2.1 OpenID 2.0 Security

A study on the security aspects of OpenID [20], highlights the danger of phishing as a major concern in OpenID. By tricking an user into giving away their OpenID authentication credentials, e.g. by setting up a fake OP, an attacker can get access to all OpenID enabled services in the name of the legitimate user. In order to do this, the attacker does not have to attack the OP directly but can set up a malicious RP which then redirects the user to the fake OP. As OPs act as Identity Providers for multiple users an attacker can use this technique to easily collect credentials for a large amount of users.

Another attack highlighted in [20] is the danger of Cross-Site-Request-Forgery (CSRF) attacks which silently logon the user to a service of the attacker's choice once the user has logged in into another RP and then allows the attacker to perform actions in the user's name. The CSRF attack relies on the fact that the OP and not the RP decides on the login security policy and that the OP does not show the login process to the user if a CSRF attack happens.

With the use of a smart card we introduce a multi-factor authentication, i.e., in order to use the smart card capabilities of our solution, the user is requested to provide an additional authentication factor such as a PIN code. As the attacker would need to be in possession of the smart card and the PIN code, phishing attacks become more difficult. Our solution is designed such that the user will be notified if a login process is taking place, preventing the issue of silent logon by CSRF attacks.

2.2 3GPP OpenID/GBA

The Generic Bootstrapping Architecture (GBA) [21] is intended to extend the security infrastructure of MNOs to applications and services on the internet. GBA provides a method for users to authenticate to services which implement a network application function (NAF) The NAF connects to the Bootstrapping Server function (BSF) of the MNO in order to bootstrap service authentication keys from the subscriber keys stored in the Home Subscriber Server (HSS).

GBA [22, p. 23-47] consists of two phases, where the first phase is a bootstrapping procedure and the second phase is an authentication phase with the NAF using the bootstrapped keys. In the case of UMTS networks the GBA keys are obtained by running the Authentication and Key Agreement (AKA) protocol [23] between the UE and the HSS with the BSF as intermediary. At the end of the bootstrapping, the BSF and UE obtain a session key Ks and a transaction identifier $B\text{-}TID$. The UE can now use this secret with the application specific NAF for authentication and securing the communication. A NAF specific key,

Ks_NAF is derived from *Ks* in the UE and the NAF gets the same *Ks_NAF* from the BSF. This establishes a shared secret between the UE and the NAF. 3GPP has investigated the option for interoperability of GBA and OpenID [16, 24, 25], providing GBA based authentication at the OP.

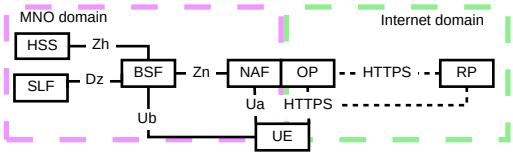

Fig. 2. Overview for an integrated OpenID/GBA Architecture according to [16]

The OpenID/GBA protocol follows the regular OpenID protocol as shown in figure 1 replacing steps 6 and 7 with GBA authentication. The solution relies on the availability of GBA functions in the network and on the device. While the network needs to implement the BSF and NAF functions, the device needs to implement a GBA module to communicate with the browser, the NAF and the UICC. The architecture of OpenID/GBA is shown in figure 2. A similar approach for integration of EAP-SIM with OpenID has been taken by [15].

2.3 SSL Certificate based OpenID Authentication

In [13, 26] a smart card based solution for OpenID authentication is presented, where a smart card stores a SSL certificate which may be used for authentication with the OP. The smart card carrying the key material and certificates is attached to a PC with a USB dongle. This solution requires an additional public key infrastructure and also requires the use of an additional USB key device. This approach is not as elegant as an approach which can leverage the already existing authentication infrastructure of MNOs.

2.4 Liberty Alliance Advanced Client

Liberty Alliance (LA) introduced in [27] a new entity, called advanced client. At the core of an advanced client is the trusted module (TM), which is an extension of the network IdP. Local applications and service providers can delegate authentication of the user to the TM instead of the network IdP. The TM is considered to run in a trusted and secure environment of the device. In [28] the deployment of a TM onto a UICC using 3GPP Over The Air (OTA) management operations [29, 30] is discussed.

3 The Smart OpenID Protocol

In order to enable an efficient use of the existing and deployed security infrastructure of MNOs and at the same time provide seamless access to service providers,

we introduce a new entity for our smart card based OpenID protocol, called local OP. The local OP, similar to the advanced client from LA [27], acts as a delegate of the network OP. The main role of the local OP is to authenticate the end user and issue the OpenID assertions to the end user's browser. The local OP therefore has an HTTP interface, which allows seamless communication with the device's browser without any modification to the browser itself. Using the capabilities provided by the UICC, the local OP can securely store secrets and perform crypto operations, e.g. creating signatures. The local OP may be implemented as a combination of a user level application, which provides the HTTP interface towards the browser and an applet on the UICC which handles all cryptographic operations and storage of keys. It is important to note that our proposed protocol does not impose any changes to the OpenID protocol, so existing RPs don't need to be modified. Due to restrictions from the mobile network, the local OP cannot be reached by RPs for the discovery and association steps of the OpenID protocol (steps 2, 3 in section 1.2, figure 1), i.e. communication to the local OP is restricted to the device only. Therefore, we introduce the OP support function (OPSF), operated by the MNO and reachable via public internet which facilitates these steps.

For the local OP to act as a delegate for the network OPSF, there needs to be a trust anchor. This trust is created by a shared secret S between the local OP running on the smart card and the OPSF. On the device, the secret is stored in the secure storage of the UICC and can only be used by the local OP. The shared secret can be established with different means, ranging from a preinstalled secret at time of personalisation of the UICC, by binding it to the network authentication or by using the MNO's OTA capabilities [29, 30]. When the user logs on to a RP using his identifier, the RP performs discovery of the OPSF and requests to establish an association. Since the RP cannot easily communicate directly with the local OP on the mobile device, the OPSF is used in this step for the discovery and association process. The OPSF looks up the shared secret S for the identifier and creates a random association handle `asc_hdl`, which is an ASCII string of at most 255 characters. The OPSF then uses a key derivation function (KDF) with input S and `asc_hdl` to create an association secret S_a, which will be valid for this OpenID session. In the association response [10, sec. 8.2], the OP returns the `asc_hdl` and the S_a to the RP, where S_a is encrypted using a Diffie-Hellman secret established between the RP and OPSF. The RP then redirects the user to the OP. Instead of following the redirect to the network OP, the browser is actually redirected to the local OP, which can be done via a modification to the local DNS lookup or by a browser plugin. The local OP then first authenticates the user, requesting for example a password or PIN code. This authentication is done locally, in contrast to the regular OpenID protocol, where user authentication credentials are sent over the internet to the OP. Local authentication not only protects from phishing or man in the middle attacks on the internet, but also introduces a two-factor authentication and allows to employ advanced authentication mechanisms, such as biometrics. Since the local OP runs on the device, it can either use a regular webpage or use the user

interface of the phone's operating system to provide a more seamless experience to the user. After the user has successfully authenticated towards the local OP, e.g. by entering a PIN code or password, the local OP checks if the identifier is in the list of allowed identifiers for this user.

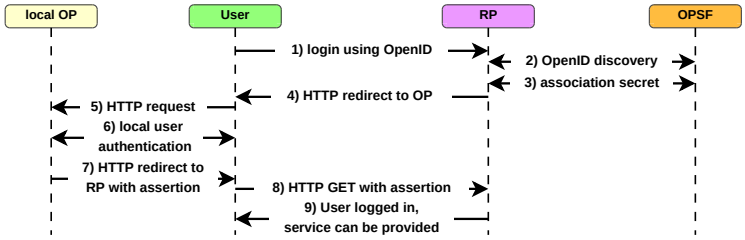

Fig. 3. Protocol Flow for the Smart OpenID protocol

If so, the local OP loads the shared secret S and together with the `asc_hdl` received from the redirected request, applies the same KDF as the OPSF to derive S_a. Using S_a, the local OP calculates the needed assertion signature and creates the redirect message containing the OpenID parameters and the signed assertion. The browser is then redirected to the RP, which can in turn verify the assertion using the S_a, as exchanged in the association with the OPSF. No additional communication is needed between the RP and OPSF.

3.1 Additional Privacy using Identifier Select Mode

In the generic description we have assumed that there is one shared secret S per local OP. This allows a single local OP to provide multiple identifiers for one user, e.g. for personal or business use. Since the security relevant operations of the local OP are in the smart card and cannot be modified by the user, the MNO has control over the identifiers that can be used by the local OP. The local OP will only create assertions for the identifiers which are enabled by the MNO. The user could for example create or buy identifiers which are then installed in the local OP if he needs separate identifiers for different purposes.

OpenID supports the so called OP-driven identifier selection [10, sec. 10], where the user only supplies the URL of its OP to the RP, e.g. `id.example`. The RP will then associate with the OP using the OP identifier `id.example` by setting `openid.identity` to `http://specs.openid.net/auth/2.0/identifier_select`. The user's browser is then again redirected to the local OP as described above. However, in this case, the local OP recognizes that identifier selection is used and displays a list of possible identifiers for the user to select between. This list is controlled by the MNO, and the user cannot select an identifier which is not in the list. After selection of an identifier, the local OP creates the signed assertion for this identifier and redirects the browser back to the RP. The RP

verifies the assertion and provides the service. As the RP does not need to communicate with the OPSF, the actual identifier used is not revealed to the OPSF and allows the creation of a privacy sensitive variant of the OpenID protocol. With a normal OpenID protocol run, the OP will always get to know the identifier at the time of user authentication. Hence, a higher level of privacy can be achieved as long as RP and OPSF do not collude.

While the use of the identifier select mode seems to have some restrictions, it can also be used for a slightly different purpose. To create unlinkability of user identifiers for the RP, a so called random id can be enabled in the local OP. Therefore the OPSF establishes a shared secret S_r with all local OP entities which have signed up for this service. The OPSF can provide a different OP URL, e.g. `r.id.example` which is then used in the identifier select mode. When the user is redirected to the local OP, the local OP allows the user to create a pseudo random identifier string, e.g. resulting in an identifier `a6gh8.r.id.example` and issue an assertion for this identifier. Since the identifiers can be created different for each login, they cannot be linked by the RP.

3.2 Attribute Assertions using Identifier Select Mode

An additional use of the identifier select mode is the combination with attributes. Upon entering a contract with the MNO, the user has provided certain identity attributes, such as name, street address, age, etc.. As an example we consider age verification, where the user has to prove that he is over 18 to access the service of the RP. The RP, e.g. an online casino, is legally bound to verify the age of its customers. Implementing an age verification system can be costly for the RP and the RP therefore relies on a statement from an entity that he trusts, i.e. the MNO. The user wishes to only reveal sufficient identity data to enable access to the service also does not want to let the MNO know that he is into online gambling. The OPSF creates group keys G_i for attribute i and shares the group key with all local OPs that are members of that group, i.e. the user's local OP has a key G_{age} installed to enable the age attribute to be provided. The user now only enters the URL of his OP at the RP, which in turn engages in an OpenID authentication using identifier select mode with the OP identifier `age.id.example`. The OPSF then derives an association secret using G_{age}. When the local OP asks the user to select an identifier, the user can select the identifier 'over18', and the local OP then creates the assertion using a key derived from the association handle and G_{age}. The RP can now be assured that the user is over 18, because the assertion is coming with the MNO trust anchor. The user did not have to enter personal data to provide proof of age to the RP and at the same time did not reveal to the MNO the access to the online casino.

3.3 Implementation

We have implemented the local OP as an Android application on a Nexus One phone and successfully demonstrated OpenID logins to different existing RPs. For the OPSF and local OP we used the Python based OpenID libraries [31]

and SL4A [32] on the Android phone. In order to demonstrate the variants described in sections 3.1 and 3.2, we have also implemented a RP demo service. Our demo implementation shows that the protocol modifications are compliant to the standards and existing RPs don't need modifications to work with the local OP. The demo is software based and does not yet include access to the

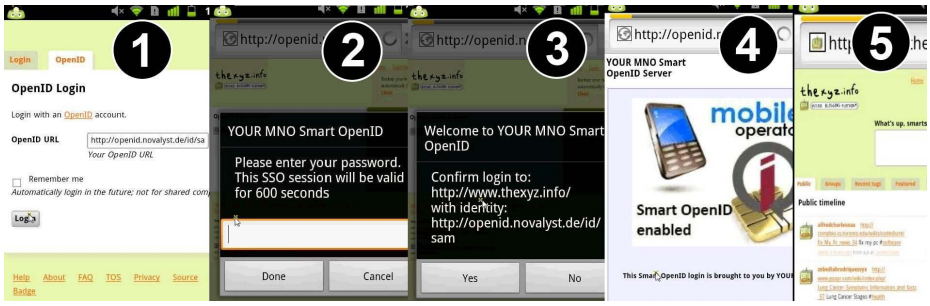

Fig. 4. Screenshots of the demo implementation, showing a login to thexyz.info

UICC, which will be added in the future. Figure 4 shows screenshots of a login process. The user enters his identifier (1) at the RP, and is then redirected to his local OP which displays a local authentication prompt, where the user is asked to provide his secret PIN code (2). After the user enters his code and confirms the login (3), a branding screen from the MNO can be displayed (4) and the service from the RP is provided (5). The user interface of the local OP is that of normal Android application creating a more seamless user experience.

3.4 Binding to Network Authentication

Instead of creating the shared secret between the local OP and the OPSF at the time of personalisation of the UICC, it is also possible to bind the use of the secret to a successful network authentication. This creates an additional level of security as the network authentication has to be successful in order for the local OP to engage in an OpenID authentication. Such a binding can for example be achieved with the GBA protocol [21]. The OPSF can be co-located with a NAF, similar to the OP/NAF combination in [16]. The local OP then runs GBA with the OPSF/NAF to get a Ks_NAF key which will then be used to derive the shared secret for the local OP and OPSF to be used with Smart OpenID.

As an alternative, there can be a combination of our proposal and the OpenID/GBA protocol [16]. When first logging in to a new RP, a full OpenID run using GBA is performed, resulting in the device and OPSF getting a Ks_NAF. Both entities then derive a RP specific K_RP key from Ks_NAF. Any subsequent login to that RP will then follow our Smart OpenID protocol and derive the association secret from the K_RP and have the local OP sign and issue the assertion. This method of binding has previously been described in [33]. A fringe

benefit of this approach is a seamless and automated login and access to the RP using the strong authentication security of the MNO. Alternative embodiments leveraging the MNO provisioned credentials such as OpenID with EAP-SIM or EAP-AKA to achieve a similar result are also feasible.

3.5 Analysis of Smart OpenID

Our solution does not require any changes to the OpenID protocol standards and remains compliant with existing RPs, which means that it can be directly deployed and enables access to a large variety of OpenID RP services. Some vulnerabilities of the standard OpenID protocol are addressed by the OpenID Security Best Practices [34]. Phishing of user credentials for an OpenID identifier should be of special concern, as OpenID enables access to all RP services. OpenID replaces multiple insecure passwords or a single password used across multiple services with one credential. Our solution increases the security by introducing a strong two-factor authentication, which is bound to the device and network authentication. Using a single point of authentication, namely the local OP, the user can easily and securely log in to different services, without having to remember different passwords. The user only has to remember his PIN code for the Smart OpenID application, which he has to provide for every authentication. This multi-factor authentication, based on possession of the card and knowledge of the PIN code, provides security from misuse in the case of a stolen or lost phone. In addition, using OTA mechanisms, the MNO can provide a portal for users, to remotely delete the smart card applet and thus prevent usage of stolen cards. The same portal could then allow users to register their new smart card with their existing identifier. Due to the local authentication, using the PIN code, credentials are not send over the internet, preventing phishing and MitM attacks on the internet. It is possible to use varying grades of multi-factor local authentication, e.g. biometrics, to further increase the security.

The local OP provides not only a secure storage of credentials, but due to its function as an assertion issuing entity, also enables attribute based access control with some level of privacy for the user. The concept allows MNOs to easily provide IdM as a service to RPs, monetizing user data that they already possess. The MNOs leverage their existing infrastructure, user data and trust to provide attributes to RPs who can in turn reduce their costs in verifying user data. The local OP, based on the UICC, further enables multi-factor authentication for OpenID. As presented, Smart OpenID can create privacy towards the network and still issue trustworthy assertions on a user's identity or attributes to RPs.

In our experiments, we successfully logged in to various RPs on the internet. We simulated additional delays that might be introduced by the processing speed of a smartcard. The operations on the smart card which are needed for Smart OpenID can be limited to the crypto operations, so that the user does not experience a delay in the authentication. Since communication to the remote server in normal OpenID is replaced by faster local communication, the user even perceives less delay using the local OP. The smart cards currently used in mobile networks are equipped with anti-temper devices and using cryptographic

protection which makes cloning of the card very difficult. Given the fact that an attacker needs physical access in order to attack the card's contents, the user has enough time to react and report the loss to his MNO to block further usage of the card. As OpenID addresses the issues of the password dilemma, the presented combination with a strong, smart card based, local authentication provides an efficient protection from attacks, and creates a secure and seamless solution for SSO. Our current research is focussed on mobile scenarios, and hence uses the UICC for the local OP. Smart OpenID can be extended to other smart cards or security tokens, such as secure micro SD cards, which are all based on the Javacard platform. The split-terminal scenario described in [16] can serve as a blueprint for the extension of the concept to other devices, such as laptops.

4 Conclusions

OpenID as a lightweight protocol for IdM is increasingly being adopted by the industry. We have shown that by introducing an additional entity, the local OP, to existing smart card security, we can create a more secure OpenID implementation without requiring changes to the protocol. In addition, we have demonstrated that the use of the identifier select in combination with the local OP creates a higher level of privacy. This concept can be seen as an *IdM Enabler* as described in [9], allowing OpenID to be used by MNOs to offer new application services, which need elevated security and rely on trusted identity information. We have implemented the local OP and the OPSF and successfully demonstrated login to different RP websites without any modification to their OpenID implementation. As UICCs are in every mobile phone, they are a prime asset of MNOs which can provide the security for the Smart OpenID protocol.

References

1. Windley, P.: Digital Identity. O'Reilly Media, Inc. (2005)
2. Liberty Alliance Project. Web page at `http://www.projectliberty.org` (2002)
3. Chappell, D., et al.: Introducing windows cardspace. MSDN. April (2006)
4. Bertocci, V., Serack, G., Baker, C.: Understanding windows cardspace Addison-Wesley Professional (2007)
5. Higgins Personal Data Service. `http://www.eclipse.org/higgins/`
6. Telco 2.0: Telco 2.0 Manifesto - Business Model Innovation for the Digital Economy. `http://www.stlpartners.com/manifesto.php`
7. Camenisch, J., Fischer-Huebner, S., Rannenberg, K.: Privacy and Identity Management for Life. Springer Verlag (2011)
8. Koschinat, S., Bal, G., Rannenberg, K.: Economic Valuation of Identity Management Enablers. PrimeLife Deliverable D6.1.2 (May 2011)
9. Koschinat, S., Bal, G., Weber, C., Rannenberg, K.: Privacy by sustainable identity management enablers. Privacy and Identity Management for Life (2011) 431–452
10. OpenID.net: OpenID Specifications. `http://openid.net/developers/specs/`
11. Uruena, M., Busquiel, C.: Analysis of a Privacy Vulnerability in the OpenID Authentication Protocol. IEEE Multimedia Communications, Services and Security (MCSS2010)(Krakow, Poland, 2010)

12. van Thanh, D., Jonvik, T., Feng, B., Van Thuan, D., Jorstad, I.: Simple strong authentication for internet applications using mobile phones. In: IEEE GLOBECOM Global Telecommunications Conference, 2008. (2008)
13. Urien, P.: Convergent identity: Seamless OpenID services for 3G dongles using SSL enabled USIM smart cards. In: Consumer Communications and Networking Conference (CCNC), 2011 IEEE, IEEE (2011) 830–831
14. Leicher, A., Schmidt, A.U., Shah, Y., Cha, I.: Trusted Computing enhanced OpenID. In: 2010 International Conference for Internet Technology and Secured Transactions (ICITST). (2010) 1–8
15. Jorstad, I., Johansen, T., Bakken, E., Eliasson, C., Fiedler, M., et al.: Releasing the potential of openid & sim. In: Intelligence in Next Generation Networks, 2009. ICIN 2009. 13th International Conference on, IEEE (2009) 1–6
16. 3GPP: Identity management and 3GPP security interworking; Identity management and Generic Authentication Architecture (GAA) interworking. TR 33.924, 3GPP (June 2011)
17. Chen, Z.: Java Card Technology for Smart Cards. Prentice Hall (2000)
18. ISO : ISO 7816-4: Identification cards - Integrated circuit cards - Organisation, security and commands for interchange (2005)
19. SIM Alliance: OpenMobile API Specification v2.0.2. `http://www.simalliance.org` (2011)
20. Tsyrklevich, E., Tsyrklevich, V.: Single Sign-On for the Internet: A Security Story. BlackHat Conference Las Vegas 2007 (2007)
21. 3GPP: 3G Security; Generic Authentication Architecture (GAA); System description. TR 33.919, 3GPP (June 2010)
22. Holtmanns, S., Niemi, V., Ginzboorg, P., Laitinen, P., Asokan, N.: Cellular Authentication for Mobile and Internet Services. Wiley (2009)
23. 3GPP: 3G security; Security architecture. TS 33.102, 3rd Generation Partnership Project (3GPP) (December 2010)
24. Weik, P., Wahle, S.: Towards a generic identity enabler for telco networks. In: Proc. 12th Internat. Conf. on Intelligence in Networks (ICIN 2008), Bordeaux. (2008) 20–23
25. Ahmed, A.S.: A User Friendly and Secure OpenID Solution for Smart Phone Platforms. Master's thesis, Aalto University, School of Science and Technology, Faculty of Information and Natural Sciences (2010)
26. Urien, P.: An OpenID provider based on SSL smart cards. In: 7th IEEE Consumer Communications and Networking Conference (CCNC). (2010)
27. Liberty Alliance: ID-WSF Advanced Client 1.0 Specifications. Technical report, (2007)
28. Liberty Alliance: ID-WSF Advanced Client Implementation and Deployment guidelines for SIM/UICC Card environment. Technical report, (2007)
29. 3GPP: Remote APDU Structure for (U)SIM Toolkit applications. TS 31.115, 3GPP (December 2009)
30. 3GPP: Remote APDU Structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications. TS 31.116, 3GPP (December 2009)
31. Janrain: Python OpenID libraries. `http://www.janrain.com/openid-enabled`
32. Scripting Layer for Android. `http://code.google.com/p/android-scripting/`
33. Schmidt, A.U., Leicher, A., Shah, Y., Cha, I.: Efficient Application SSO for Evolved Mobile Networks. In: Proceedings of the Wireless World Research Forum Meeting 25 (WWRF 25), London, UK, 2010
34. OpenID Foundation: OpenID security best practices. `http://wiki.openid.net/OpenID-Security-Best-Practices`