Addressing privacy issues in CardSpace

Waleed A. Alrodhan and Chris J. Mitchell
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom
{W.A.Alrodhan, C.Mitchell}@rhul.ac.uk

Abstract

CardSpace (formerly known as InfoCard) is a Digital Identity Management system that has recently been adopted by Microsoft. In this paper we identify two security flaws in CardSpace that may lead to a serious privacy violation. The first flaw is the reliance on Internet user judgements of the trustworthiness of service providers, and the second is the reliance of the system on a single layer of authentication. We also propose a solution designed to address both flaws. Our solution is compatible with the currently deployed CardSpace identity metasystem, and should enhance the privacy of the system with minor changes to the current CardSpace framework. We also provide a security and performance analysis of the proposed solution.

1 Introduction

Along with the growing reliance on Internet web applications in our daily life, comes the problem of managing the necessary digital identities and preserving their privacy. In an open large-scale domain such as the Internet, preserving user privacy is not a straightforward task. Identity theft, which occurs when an impostor uses a legitimate user's identifying information without his/her consent, is becoming one of the biggest concerns for organisations offering services on the Internet.

Many solutions have been proposed in the last few years to address the threat of identity theft, and to tackle identity-oriented attacks such as phishing and pharming. Most of those solutions are based on the concept of *Identity Federation* (different identities that belong to the same user in a particular trust domain are "federated"), and *Single Sign-On* (where a user performs an authentication process only once in a single working session).

Recently, Microsoft has proposed a new identity management framework named CardSpace. CardSpace has some similarities to other identity federation systems; however it is not a single sign-on system. CardSpace is designed to reduce the reliance on passwords for Internet user authentication by service providers, and to improve the privacy of personal information.

In this paper we identify significant security and privacy issues in the CardSpace scheme. We focus on two particular security problems, namely the reliance by the system on Internet user judgements of the trustworthiness of service providers, and the dependency on a single layer of user authentication with the Identity Provider. In this paper we propose a solution for these two problems, using the concept of Secured from Identity Theft (SIT) attributes [2], and zero-knowledge cryptographic techniques.

The remainder of this paper is organised as follows. In section 2 we provide a brief overview of the CardSpace framework. In section 3 we describe two security flaws in CardSpace. In section 4 we propose a solution for the security problems discussed in section 3, and in section 5 a security and performance analysis of the proposed solution is given. Section 6 concludes the paper.

2 Microsoft CardSpace

In this section we provide a brief overview of CardSpace. We then describe the CardSpace framework and message flow.

2.1 An Overview

CardSpace is the name for a Microsoft WinFX set of software components that form an identity management system or an *identity metasystem*, since it is a system of systems. This identity metasystem is designed to comply with the Laws of Identity promulgated by Microsoft[1].

Digital identities in CardSpace are represented as claims made by one digital subject (e.g. an Internet user) about itself or another digital subject. A claim is an assertion that certain identifying information (e.g. given name, SSN, credit card number, etc.) belongs to a given digital subject [3]. According to this definition, identifiers (e.g. username)

and attributes (e.g. user gender) are both treated as claims within the identity metasystem.

CardSpace can be integrated with Microsoft Windows XP and Internet Explorer version 7 (a toolkit is freely available from Microsoft), and has been distributed with Windows Vista. Since CardSpace is an "open" XML-based framework, CardSpace plug-ins for browsers other than Microsoft Internet Explorer can also be developed, such as the Firefox Plug-in.[2]

2.2 The CardSpace Framework

The CardSpace framework is based on the identification process we experience in the real world using physical identification cards. Within the CardSpace framework, an identity provider issues Internet users with virtual cards called *InfoCards*, that hold non-sensitive meta-information related to them. Subsequently, the Internet users can use their InfoCards to identify themselves to any service provider (or relying party) who trusts the identity provider that issued the InfoCards. InfoCards can also be self-issued by the Internet users themselves.

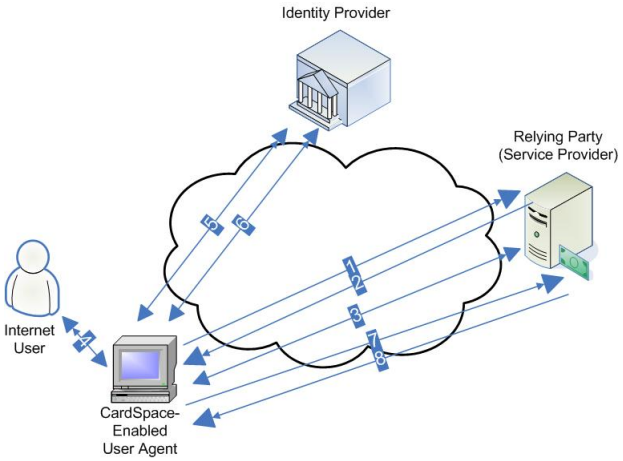

Figure 1. CardSpace Framework.

Figure 2 provides a simplified sketch of the CardSpace framework. In the figure it is assumed that the user has already been issued an InfoCard by the identity provider (henceforth abbreviated to IdP). In step 1, the CardSpace-enabled user agent or the *Service Requestor* (henceforth abbreviated to CEUA), which is essentially a CardSpace-enabled web browser, requests a service from the relying party or service provider (henceforth abbreviated to RP). In step 2, the RP identifies itself using a public key certificate (e.g. a certificate used for SSL/TLS), and declares itself as a CardSpace-enabled RP using XHTML code or HTML object tags. After recognising that the RP is CardSpace-enabled, the CEUA retrieves the RP security policy in step 3. This policy contains a list of the claims that must be asserted about the Internet user (henceforth abbreviated to user) in order for this user to be granted the service, the IdPs that are trusted to make such assertions, and the types of security token holding the assertions that are acceptable to the RP. The security policy also specifies requirements that must be met by the retrieved security token (e.g. the type of proof key and the maximum token age). It is important to emphasise here that CardSpace identity metasystem does not demand specific types of tokens; any token type can be used within the framework.

In step 4 the CEUA matches the RP's security policy with the InfoCards possessed by the user, in order to find an InfoCard that satisfies the RP's policy. If one or more suitable InfoCards are found, the user is prompted to select an InfoCard from amongst them. After the user has selected an InfoCard, the CEUA initiates a connection with the IdP that issued that InfoCard. The user performs an authentication process with the IdP in step 5.

If the authentication process succeeds, step 6 takes place, in which the CEUA requests the IdP to provide a security token that holds an assertion of the truth of the claims listed within the selected InfoCard; the message that holds this request is called a *request security token* (or RST) message. The IdP will then check whether its security policy permits it to generate the requested security token. If so, the IdP will reply by sending a security token within a message called a *request security token response* (or RSTR). Finally, the CEUA forwards the security token to the RP in step 7, and, if the RP verifies it successfully, the service will be granted in step 8.

It is worth mentioning here that, after step 6, the security token can be optionally displayed to the user before proceeding to step 7. Moreover, the RP will get an assertion from the IdP that the security token received was issued to a particular user. This assertion is based on the concept of "proof-key" where the user proves that she/he is in possession of the proof-key held in the security token. This assertion helps to prevent token replay attacks.

The *Security Token Service* (henceforth abbreviated to STS) is a software component of the CardSpace identity metasystem, responsible for security policy and token management within the IdP (and optionally within the RP).

The CardSpace identity metasystem makes use of XML-based protocols, including the Web Services (WS-*) protocols and SOAP. Most of these protocols require the RP to have an STS server in order to process the messages [1, 9]. The message flows of the CardSpace framework are as follows:

1. **CEUA** → **RP** : HTTP GET Login HTML Page Request

[1] http://msdn2.microsoft.com/en-us/netframework/aa663320.aspx
[2] http://xmldap.blogspot.com/2006/05/firefox-identity-selector.html

2. **RP** → **CEUA** : HTML Login Page + InfoCard Tags (XHTML or HTML object tags)
3. **CEUA** ↔ **RP** : CEUA retrieves security policy via *WS-SecurityPolicy*
4. **CEUA** ↔ **User** : User picks an InfoCard
5. **CEUA** ↔ **IdP** : User Authentication
6. **CEUA** ↔ **IdP** : CEUA retrieves security token via *WS-MetadataExchange* and *WS-Trust*
7. **CEUA** → **RP** : CEUA presents the security token via *WS-Trust*
8. **RP** → **CEUA** : Welcome, you are now logged in!

WS-MetadataExchange [4], WS-Trust [6] and WS-SecurityPolicy [7] messages are transported over SOAP. Messages in steps 3, 5, 6 and 7 must be carried over an SSL/TLS channel to preserve their confidentiality. If the RP does not have an STS server, the messages in steps 3 and 7 will be carried using HTTP over an SSL/TLS channel. It appears reasonable to assume that the most commonly used security token type is a SAML assertion, carried over SOAP. Integrity of the security token is preserved using an XML-Signature as part of the WS-Security [9] protocol.

3 Security Limitations of the CardSpace Framework

We next discuss certain security limitations of the CardSpace framework. CardSpace suffers from a number of such limitations, such as its reliance on DNS names to identify the IdPs and the RPs. If the DNS server is controlled by an attacker, it can direct the identity metasystem parties to false websites. This problem is common to most of the current Internet identity management solutions. Another limitation is that, in the default scenario for the CardSpace framework, the IdP is aware of the identities of the RPs to which the user attempts to log in. Accordingly, the IdP can learn about the behaviour of users on the web. Although there is an alternative scenario, we believe that this is a potentially serious privacy violation.

In the remainder of this section we focus on two particular security limitations of the CardSpace framework which we believe are most significant, namely: the reliance on the user's judgement of the trustworthiness of the RP, and the reliance on a single layer of authentication.

3.1 Judgements of RP Trustworthiness

The user judgement regarding the honesty of the RP is a security-critical task, as stated earlier in this paper. As described in section 2.2, the RP will obtain personal information belonging to the user in the form of "asserted claims" within a security token, as sent in step 7 of the message flow. That means that, if the RP is not trustworthy, it could gather information about users and potentially use this in unauthorised ways. Accordingly, any misjudgement of the trustworthiness of an RP could result in a serious privacy violation. Hence, the task of judging the honesty of the RP is a very important one.

Within the deployed CardSpace framework, as described in section 2.2, when the user is prompted for its consent to be authenticated to an RP using a particular InfoCard, the user makes a judgement regarding the trustworthiness of the RP based on one of:

1. A high-assurance public key certificate belonging to the RP,
2. An "ordinary" public key certificate belonging to the RP (e.g. a certificate used for SSL/TLS), or
3. No certificate at all [3].

Microsoft recommends the first option, i.e. the use of a high assurance certificate (also referred to as a "higher-value" or "higher-assurance" certificate). Such a certificate is an X.509 certificate that is only issued after a rigorous and well-defined registration process, unlike the CA-specific procedures used for issuing certificates commonly employed as the basis for SSL/TLS security. A high assurance certificate includes a digitally signed bitmap of the RP's company logo, in order to make it easier for the user to identify the certificate holder.

In general, it would appear that an user is not qualified to make such a security critical decision. Most users do not pay much attention when they are asked to approve a digital certificate, either because they do not understand the importance of the approval decision, or because they know that they must approve the certificate in order to get access to a particular website. RPs with no certificates can be used in the CardSpace framework (given user consent), and this leads to a serious risk of a privacy violation. If we consider the potentially massive number of RPs, it is likely that (at least initially) many of them will not possess a high assurance certificate. Even in the case where an RP does have a high assurance certificate, the user may be deceived by a company name or logo that is similar to that used by a legitimate RP (although in principle this should be prevented by the registration process for a high assurance certificate).

It is important to emphasise that this problem is less critical in other deployed identity management frameworks such as the Liberty Alliance Project[3] and OpenID[4]. In the Liberty Alliance framework, no personal information is revealed to the service provider (or the RP); the RP gets only an assertion from the IdP that a particular user has been authenticated using a specific authentication method. The only framework-related problem arising from trusting an imposter RP within the Liberty Alliance framework would

be revealing information about the existence of a relationship between an user and a certain IdP [11].

3.2 Reliance on a Single Layer of Authentication

As discussed in section 2, the security of the CardSpace identity metasystem relies on the authentication of the user by the IdP. In a case where a single IdP and multiple RPs are involved in a single working session, which we expect to be a typical scenario, the security of the identity metasystem within that working session will rely on a single layer of authentication. This user authentication can be achieved in several ways (e.g. using an X.509 certificate, Kerberos v5 ticket, self-issued token or password); however, it seems likely that, in the majority of cases, a simple username/password authentication technique will be used.

If a working session is hijacked (e.g. by compromising a self-issued token), or the password is cracked (e.g. via guessing, brute-force, key logging, or dictionary attacks), the security of the whole system will be compromised. It is fair to mention here that most of the deployed Internet identity management solutions, such as Liberty and OpenID, suffer from the same vulnerability.

4 Improving the Security of CardSpace

Our proposed solution is based on the concept of Secured from Identity Theft (SIT) attributes [2], which is based on Schnorr's zero-knowledge protocol [5, 10]. We treat the claims within the CardSpace framework as SIT attributes. The goal of the solution is to prevent the need to reveal the actual values of the claims to any party within the CardSpace framework. This means that no party will have to trust any other party to the level that it has to reveal the actual values of the claims to it.

In our proposed solution, instead of including the actual value of the claim in the security token in step 6 of the message flow illustrated in section 2.2, the IdP will include data computed using the actual value of the claim. It must not be feasible for the RP to deduce the value of the actual claim using only this data. It merits mentioning here that the structure and the content of the security token will remain the same (e.g. time-stamps, pseudonyms, signature values, etc.), except the part that includes the actual value of the claim.

4.1 Protocol Requirements

Prior to use of the protocol, the Identity Provider must select three domain parameters, p, q and g, where p and q are large primes satisfying $q|(p-1)$, and g is an element of multiplicative order q in $\mathbb{Z}_p^*$. These domain parameters must be made known to the CEUA and RP in a reliable way, e.g. by inclusion in a certificate signed by a trusted CA. The CEUA and RP are required to know the actual value of the claim prior to the protocol run, or at least know that it lies within a small set of possible values (this can be achieved by imposing a registration procedure between the user and the RP prior to the protocol, whereby the user registers her/his claim values that can later be asserted to this particular RP).

4.2 Protocol Steps

The following protocol (see [5, 10]) forms the basis of the proposed solution.

1. **IdP** $\rightarrow$ **CEUA** : $s = g^{-c} \bmod p$ [where c is the claim value, and s is included in a security token].
2. **CEUA** $\rightarrow$ **RP** : s, $d = g^r \bmod p$ [where r is a random integer $(1 \leqslant r \leqslant q-1)$ chosen by the CEUA].
3. **RP** $\rightarrow$ **CEUA** : e [e is a random integer $(1 \leqslant e \leqslant 2^t)$ chosen by the RP, and t is a security parameter].
4. **CEUA** $\rightarrow$ **RP** : $y = r + ec \bmod q$
5. **RP**: if $d = g^y s^e \bmod p$, then user authentication is successful.

All the messages sent in the protocol above must be conveyed over a secure channel that protects both confidentiality and integrity (e.g. an SSL/TLS channel). The protocol can easily be integrated with the currently deployed CardSpace identity metasystem; no changes to the metasystem are required. However, some minor changes must be made to the framework and the way each party handles the security tokens. Steps 1, 2 and 5 of the above protocol should be integrated with steps 6, 7 and 8 respectively of the message flow described in section 2.2. The value s will be digitally signed by the IdP by including it within the security token (e.g. using an XML-signature within a SAML assertion).

After the second step of the protocol above, the RP knows that the IdP is asserting a claim c, from the inclusion of $s = g^{-c} \bmod p$ in the token; if, moreover, the RP knows in advance the expected value of c, then it can use the received value s to verify whether the IdP is asserting this expected value or not. Also, if the RP knows that c lies within a certain small set of values, then the RP can determine which is being asserted by a simple trial and error process; however, if the set of possible values for c is very large, then the RP does not learn anything about the asserted claim. After the protocol has completed, and if user authentication is successful, then the RP can grant the service to the user. Not only does successful completion of the protocol mean that the IdP is asserting the claim regarding the

[2] http://www.projectliberty.org
[3] http://www.openid.net

user, but it also proves that the user knows the claim value c, providing an additional layer of user authentication. Of course, how strong this will be will depend on whether the claim is readily guessable by a third party.

The protocol thus enables the IdP to assert a claim about the user, and for the user to confirm knowledge of this claim, without revealing the claim to the RP. This means that the user does not need to trust the RP not to misuse a revealed claim. Also note that the scheme has the advantage that it does not require any additional key management.

In the case of self-issued tokens, there is no IdP in the framework. The user must include the value $s = g^{-c} \bmod p$ instead of the actual value of the claim in the security token.

5 Analysis

In this section we provide a security and performance analysis of the proposed solution.

5.1 Security Analysis

We believe that the above solution will enhance the overall security of the currently deployed CardSpace identity metasystem. In this section we consider the security properties of the proposed scheme.

5.1.1 Addressing the CardSpace Security Limitations

In section 3 we discussed certain security limitations of the CardSpace framework. In particular, we highlighted its reliance on the user's judgement of the trustworthiness of the RP, and on a single layer of authentication. We believe that the proposed solution addresses both these security limitations.

The scheme avoids the need to rely on the user's judgement of the trustworthiness of the RP, by avoiding the need for initial trust between the user and the RP. In the revised protocol, the user does not have to reveal personal information to the RP. Instead, the user demonstrates knowledge of this information.

Our solution does not rely on a single layer of authentication. If the working session is hijacked (e.g. by compromising a self-issued token), or the user's password is cracked, the security of the whole system will not be totally breached, since the solution adds a new layer of authentication. When trying to log-in to an RP, an attacker will not be able to demonstrate the knowledge of the legitimate user's personal information, and hence that RP will not let the attacker log in. Moreover, the attacker cannot learn the legitimate user's personal information, since the actual value of the claims will not be included in the security token.

5.1.2 Privacy

We believe that the proposed solution should increase the privacy level of CardSpace users. As shown in section 4, the values of claims are not revealed at any stage. Avoiding the need to reveal the values of the claims is a significant enhancement to the privacy of CardSpace.

Unlike the currently deployed CardSpace identity metasystem, in the proposed solution the user does not have to reveal to the IdP the identity of the RP. This should also enhance the privacy of the users.

The solution implicitly assumes that the number of possible values for a claim c is greater than 2^{128}. As a result, it should be computationally infeasible for polynomially-bounded adversary to deduce the value of c from the value s (assuming that the Discrete Logarithm problem is difficult [8]).

The proposed solution satisfies the requirements of law 2 of Microsoft's own laws of identity to a greater degree than the currently deployed CardSpace identity metasystem, where law 2 states that only the minimum amount of identifying information must be revealed.

5.1.3 The Guessing Problem

Since that the proposed solution is based on disguising the personal information of the users, there is always the risk of an attacker guessing this information and breaking the second layer of authentication the scheme provides. Some claims can be guessed easily, especially for "user-oriented" attacks where information about the user is already known by the attacker. Examples of such claims include first name, home country, age or marital status. In the proposed solution, if an attacker successfully broke the CardSpace first layer of authentication (which might, for example, be password-based), then she/he can try to guess a particular claim, and verify whether her/his guess is correct or not even before forwarding the security token to the RP. This can be done using the publicly known parameters p and g, and the value s received at step 1 of the protocol-run.

We propose two solutions for this problem. The first solution, which we recommend, is based on choosing "hard-to-guess" claims by the RP to be asserted by the IdP, such as a combination of a series of attributes, such as mother's maiden name, social security number and credit card number. Since the impact of a successful guessing attack would be allowing an imposter user to log in to an RP, the RP could protect itself by requesting claims that cannot be easily guessed. Many Internet service providers already rely on "hard-to-guess" personal information to authenticate users when they forget their passwords.

Another solution would be for the IdP to mask the value c, e.g. by using the value $c + x$ instead of c, where x is a random value selected by the IdP. The value of x can then

be shared with the RP by encrypting it using the RP public key and inserting it into the security token. This solution requires the user to reveal the identity of the RP to the IdP, and this potentially violates user privacy.

Finally, there is a risk of a fake RP guessing the personal information of the user and verifying the correctness of its guesses using the publicly known parameters and the value s. The first solution described above addresses this problem; the user can refuse to request an assertion for claims that can be easily guessed.

5.1.4 Access to Claims by the CEUA

The proposed solution requires the CEUA to be aware of the actual value of the claim in order to generate a response message for the zero knowledge challenge message it receives from the RP. In some cases it is not realistic to expect users to memorise all of their registered claims values, so that they can pass them to the CEUA when required. Certain claims can be hard to remember, such as a health record number or a credit card number. Moreover, being required to enter the actual values of the claims every time a user logs in to a website might be extremely inconvenient.

We propose three solutions for this problem, although each solution has certain drawbacks:

1. *Storing the claims values on a trusted server*: from which the users can retrieve all of their registered claims after being authenticated. This solution would add more complexity to the framework.
2. *Storing the claims values on a user token*: Such a solution is potentially more reliable and less complex than the first solution. Storing the claims on a user token, such as USB memory stick or smart card, would add an authentication factor to the scheme, i.e. the possession of the token. This solution is similar to the ID card identification process used in the real world, where a person needs to present an identification card in order to be authenticated. The management and security of the token is an issue here.
3. *Retrieving the claims values directly from the IdP by the CEUA*: after authenticating the user, prior to the step of requesting a security token. Such a process would have to take place outside of the current CardSpace framework. Applying this solution means adding one more message to the framework and losing the additional layer of authentication.

5.2 Performance Analysis

The proposed solution can readily be integrated with the currently deployed CardSpace identity metasystem. Only two steps need to be added to the framework described in section 2; these two steps involve exchanging the zero-knowledge-proof messages and should take place at the end of the message flow. An additional step may be added when adopting the third proposed solution to the problem of retrieving the claims values by the CEUA.

The proposed solution requires some minor changes to the content of the security token, involving some inexpensive computational operations (i.e. performing the calculations described in section 4.2 in the protocol-run). Apart from that, the metasystem remains precisely the same (e.g. the security token format, the message flow, etc.).

The shared parameters p, q and g can be changed frequently if required, and the task of deploying these shared parameters among the involved parties can be achieved using a number of simple methods. One method would be to publish these parameters on the IdP website.

6 Conclusion

In this paper we have provided an overview of the CardSpace identity metasystem framework, and outlined certain security limitations of this framework. We focused on two security limitations, namely the reliance on the user's judgement on the trustworthiness of the RP, and the reliance on a single layer of authentication.

We have proposed a solution to address these two security limitations. The proposed solution is based on applying the concept of Secured from Identity Theft (SIT) attributes, based on Schnorr's zero-knowledge protocol, to the CardSpace identity metasystem framework. The proposed solution may be vulnerable to guessing attacks; however, we have proposed a variety of measures to mitigate the risk of such attacks.

Finally, the proposed solution can readily be integrated into the currently deployed CardSpace identity metasystem, and only two (or three) steps need to be added to the framework. The proposed solution requires some minor changes to the content of the security token issued by the IdP, with a few inexpensive computations to be performed by the involved parties.

References

[1] K. Beznosov, D. J. Flinn, S. Kawamoto, and B. Hartman. Introduction to Web services and their security. *Information Security Technical Report*, 10:2–14, 2005.

[2] A. Bhargav-Spantzel, A. C. Squicciarini, and E. Bertino. Establishing and protecting digital identity in federation systems. In *Proceedings of the 2005 ACM Workshop on Digital Identity Management, Fairfax, Virginia, USA*, pages 11–19. ACM, November 2005.

[3] K. Cameron and M. B. Jones. Design rationale behind the identity metasystem architecture, February 2006. Microsoft Corporation.

[4] F. Curbera, S. Parastatidis, and J. Schlimmer (editors). Web services metadata exchange (WS-MetadataExchange) — version 1.1, August 2006. BEA Systems, Computer Associates, IBM, Microsoft, SAP AG, Sun Microsystems, and webMethods.

[5] U. Fiege, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC: Proceedings of the nineteenth annual ACM conference on Theory of computing, New York, NY, USA*, pages 210–217. ACM, 1987.

[6] M. Gudgin and A. Nadalin (editors). Web services trust language (WS-Trust), February 2005. IBM, Microsoft and Actional, BEA, Computer Associates, Layer 7, Oblix, OpenNetwork, Ping Identity, Reactivity, and Verisign.

[7] C. Kaler and A. Nadalin (editors). Web services security policy language (WS-SecurityPolicy), July 2005. International Business Machines Corporation, Microsoft Corporation, RSA Security Inc., and VeriSign Inc.

[8] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001.

[9] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker (editors). Web Services Security: SOAP message security — version 1.1, February 2006. OASIS Standard Specification, OASIS Open.

[10] C. P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *Advances in Cryptology – CRYPTO 89: Proceedings of the ninth Annual International Cryptology Conference, Santa Barbara, California, USA*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1990.

[11] T. Wason (editor). Liberty ID-FF architecture overview — version: 1.2, 2003. Liberty Alliance Project.