Improving the Security of CardSpace

Waleed A. Alrodhan and Chris J. Mitchell
Information Security Group
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 shortcomings in CardSpace that could lead to a serious privacy violation. The first is its reliance on user judgements of the trustworthiness of service providers, and the second is its reliance on a single layer of authentication. We also propose a modification designed to address both flaws. The proposed approach is compatible with the currently deployed CardSpace identity metasystem, and should enhance the privacy of the system whilst involving only minor changes to the current CardSpace framework. We also provide a security and performance analysis of the proposal.

1 Introduction

The growing use of Internet web applications gives rise to 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 security concerns both for users and for organisations offering services on the Internet.

Many solutions to the threat of identity theft, and to tackle identity-oriented attacks such as phishing and pharming, have been proposed in the last few years. One class of solutions is based on the notions of *Identity Federation*, where different identities for the same user in a particular trust domain are 'federated', and *Single Sign-On*, where a user only needs to authenticate once in a single working session.

In 1999, Microsoft adopted .NET Passport, an identity federation and ticket-based single sign-on system. Although .NET Passport was supported by a number of well-known service providers, such as eBay and Visa, it was not widely used for single sign-on. The single sign-on features have since

been dropped, and Passport now functions simply as a means of logging into Microsoft sites. In 2005, Microsoft published two papers that discuss the "failure" of .NET Passport [5, 12].

More recently, Microsoft proposed a new identity management framework called 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 its reliance on user judgements of the trustworthiness of service providers, and its dependency on a single layer of user authentication to the Identity Provider. In this paper we propose a solution for these two problems, using the concept of Secured from Identity Theft (SIT) attributes [3] 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. In section 5 a security and performance analysis of the proposed solution is given, and in section 6 other possible solutions are briefly discussed. Section 7 concludes the paper.

A preliminary version of the solution described in section 4 was given in [1], although the analysis in section 5 extends considerably that given there. Moreover, the alternative solutions given in section 6 have not previously been published.

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 [5], promulgated by Microsoft.

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 [6, 13]. Under this definition, user identifiers (e.g. a username) and user 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 be developed, such as the Firefox Plug-in[1].

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 a user with a virtual card called *InfoCard*, which is an XML file containing (relatively) non-sensitive meta-information about the user. Subsequently, a user can use one of its InfoCards to help identify itself to any service provider who trusts the identity provider that issued the selected InfoCard. InfoCards can also be self-issued by the users themselves.

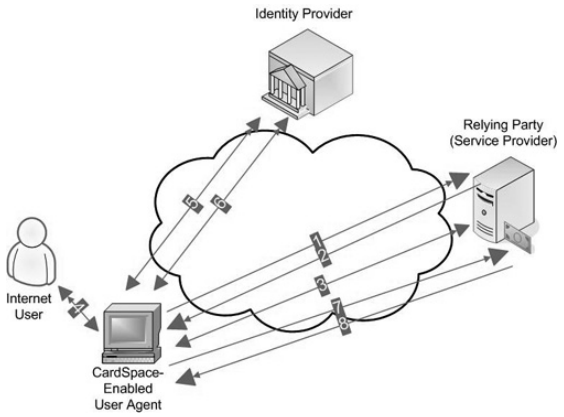

Figure 1: CardSpace Framework.

Figure 1 provides a simplified sketch of the CardSpace framework. In the figure it is assumed that the user has already been issued an InfoCard by an identity provider (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 (RP), i.e. the service provider. 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 claim types 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

[1]http://xmldap.blogspot.com/2006/05/firefox-identity-selector.html

trusted to make such assertions, and the types of security token 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, or the maximum token age). It is important to emphasise here that CardSpace identity metasystem itself does not restrict the type of security tokens, i.e. all types of token 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 one 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* 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* message. 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 contents of the security token can optionally be 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 use of a secret 'proof-key', where a user asserts ownership of a security token by demonstrating knowledge of the proof-key included in the token [14]. This assertion helps to prevent token replay attacks, i.e. where an attacker 'steals' a token for another user. The RP can select one of two types of proof-key:

1. *Symmetric* (default): In this case, the CEUA must reveal the identity of the RP to the IdP. The IdP then generates a secret key, encrypts it with the RP's public key, and inserts it inside the security token. This secret key is sent to the CEUA in step 6 (over an SSL/TLS channel). The CEUA can now use this secret key to prove rightful possession of the security token.
2. *Asymmetric* (recommended by Microsoft): In this case, the CEUA generates an ephemeral RSA key pair, and sends the public key to the IdP. The IdP then inserts this public key inside the security token. The CEUA can now use the corresponding private key to prove rightful possession of the security token.

The CardSpace identity metasystem makes use of XML-based protocols, including the Web Services (WS-*) protocols and SOAP. The message flows of the CardSpace framework are as follows:

1. **CEUA** → **RP** : HTTP GET Login HTML Page Request
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 [7], WS-Trust [9] and WS-SecurityPolicy [10] messages are transported over SOAP. The messages in steps 3, 5, 6 and 7 must be carried over an SSL/TLS channel to preserve their confidentiality. It appears reasonable to assume that the most commonly used security token type will be a SAML assertion, carried over SOAP. The integrity of the security token is preserved using an XML-Signature as part of the WS-Security [15] protocol.

3 Security Limitations of CardSpace

We next discuss certain security limitations of the CardSpace framework. One such limitation is 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 many current Internet identity management solutions, and is very difficult to address. Probably the only long term solution to this problem is to hope that the use of DNSSEC [2], or some other secure address resolution solution, will become widespread.

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. One solution to this issue is to hope that, at some future time, use of the CardSpace option in which RPs remain anonymous becomes the norm.

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

addressed here partly because of their potential seriousness, and partly because, unlike other issues, no solution appears to be in place, even for the long term.

3.1 Judgements of RP Trustworthiness

The user judgement regarding the honesty of the RP is a security-critical task. 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. Thus, if the RP is not trustworthy, it could gather information about users and potentially use this information 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.

In the 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.

Obviously, in the third situation the user has no evidence of the honesty of the RP [6].

Microsoft recommends the first option, i.e. the use of a high assurance certificate [4, 14] (also referred to as a 'higher-value', 'higher-assurance' or 'extended validation' 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 might include a digitally signed bitmap of the RP's company logo, in order to make it easier for the user to identify the certificate holder[2]. Figure 2 shows an example of a CardSpace message to the user describing a high assurance certificate issued by 'Verisign' to a company called 'Overdue Media'. The 'check mark' beside a certificate's field is an indication that the certificate issuer has assurance of the veracity of that field.

In general, it would appear that a typical user is not qualified to make such a security critical decision. Most users do not pay much attention

[2]The inclusion of such a logo is discussed in a number of documents circulated by Microsoft [6, 14], although the latest version of the draft standard for extended validation certificates [4], as published by the CA/Browser Forum, does not mandate the inclusion of a logo. Whether or not such a requirement will be included in the standard at a later date remains unclear.

Figure 2: An example of a higher-value certificate [6].

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 without any certificates at all 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 and the user is careful, 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 some other identity management frameworks, such as those of 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 a user and a certain IdP [17]. That is, the problem arises in CardSpace because CardSpace allows claims about users to be handled in a transparent way. These claims could contain extremely sensitive user information, and a user may not be aware what user information is being passed to the RP (even though the CardSpace interface may provide a list).

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

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, i.e. the authentication of the user to the IdP. This user authentication can be achieved in a variety of 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 entire 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

In this section we describe a scheme designed to address both of the major security limitations discussed in section 3, (i.e. reliance on a user judgement of RP trustworthiness and a single layer of authentication). It is based on the concept of Secured from Identity Theft (SIT) attributes [3], which is based on Schnorr's zero-knowledge protocol [8, 16]. We treat the claims within the CardSpace framework as SIT attributes.

The goal 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.

The scheme operates as follows. 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 value of the claim. It must not be feasible for the CEUA or the RP to deduce the value of the 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. We now examine the operation of the scheme in greater detail.

It is important to note that this scheme is not intended to replace the use of extended validation certificates. Such certificates, particularly if they include a readily recognisable item such as a corporate logo, are a potentially valuable way of improving user understanding of authentication issues.

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 IdP, CEUA and RP are all 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 requiring the user and the RP to conduct a registration procedure prior to use of the protocol, in which the user registers the claim values that can later be asserted to this particular RP).

4.2 Protocol Steps

The following protocol (see [8, 16]) 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 above protocol must be conveyed over a 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; indeed, no changes to the metasystem are required. However, some minor changes must be made to the framework and the way that 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 should 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, 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 user, but it also proves that the user knows the claim value c, providing an additional layer of user authentication. Of course, the strength of this additional layer of authentication 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.

The above scheme is based on a specific cryptosystem, namely the Schnorr protocol. It would be possible to replace this scheme with any other scheme with similar properties. More specifically, we require a protocol which enables both the CEUA and the RP to prove knowledge of the a claim c, so that the RP knows that they both know the claim, but so that the data is not revealed in this process. The particular advantage of the Schnorr scheme in this context is that achieves all the objectives in an efficient way, and can be seamlessly incorporated into the CardSpace message flows.

5 Analysis

We now provide a security and performance analysis of the scheme.

5.1 Security

We first consider the scheme's security properties.

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 scheme proposed in section 4 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 trust between the user and the RP. In the revised protocol the user does not reveal personal information to the RP. Instead, the user demonstrates knowledge of this information. Of course, the user will still have to trust the RP at least once, in order to

register her/his personal information with that RP (e.g. when he/she first registers with that RP), and this trust is likely to be based on a public key certificate (e.g. the RP's SSL certificate). However, it appears reasonable to assume that the user will be more careful during this one-off registration procedure than in routine use of the RP service.

The modified protocol no longer relies 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 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 knowledge of the user's personal information, and hence the RP will not let the attacker log in. Moreover, the attacker cannot learn the user's personal information, since the claim values will not be included in the security token.

5.1.2 Privacy

We believe that the scheme should increase the privacy level of CardSpace users. As shown in section 4, claim values are not revealed at any stage. This is a significant enhancement to the privacy of CardSpace.

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

The scheme implicitly assumes that the number of possible values for a claim c is greater than 2^{128} (see also section 5.1.3 below). As a result, it should be computationally infeasible for an adversary to deduce the value of c from the value s (assuming that the Discrete Logarithm problem is difficult [11]).

The proposed scheme 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 this law states that only the minimum amount of identifying information must be revealed.

5.1.3 The Guessing Problem

Since the scheme is based on disguising user personal information, there is always the risk of an attacker guessing this information and breaking the second layer of authentication that the scheme provides. Some claims can easily be guessed, 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, and marital status. If an attacker successfully broke the CardSpace first layer of authentication (which might, for example, be password-based), then she/he could try to guess a particular claim and verify whether or not her/his guess is correct 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 in step 1 of the protocol run.

We propose two possible solutions for this problem. The first, which we recommend, requires the RP to choose 'hard to guess' claims 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 a successful guessing attack, if combined with breaking the first level of authentication, would allow an imposter user to log in to an RP, the RP could protect itself by requesting claims that cannot easily be guessed. Indeed, many Internet service providers already rely on 'hard to guess' personal information to authenticate users when they forget their passwords.

Another approach 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. However, this solution requires the user to reveal the identity of the RP to the IdP, and this removes one of the advantages of the scheme (as discussed in section 5.1.2).

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/CEUA 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 to the challenge message it receives from the RP. In some cases it is not realistic to expect users to memorise all of their registered claim 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. Hence, we consider other ways by which the CEUA can retrieve the actual values of the claims.

We propose three approaches, although each has certain limitations:

1. *Storing the claim values on a trusted server*: users could retrieve their registered claim values from a third party server (after being authenticated by the server). However, this would add complexity to the framework.
2. *Storing the claim values on a hardware user token*: Such an approach is potentially more reliable and less complex than the first approach. Storing the claims on a hardware 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. However, token provisioning and management adds significant user complexity.

3. *Retrieving the claim values directly from the IdP by the CEUA*: the user could request the IdP to provide it with the claim values, after the IdP has authenticated the user, and prior to requesting a security token. Such a process would have to take place outside the current CardSpace framework. This would mean adding one more message to the framework and losing the additional layer of authentication.

5.2 Performance

The proposed scheme 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 needed if the third proposed solution to the problem of retrieving the claims values by the CEUA is adopted (as described in section 5.1.4).

Incorporating the protocol into the CardSpace message flow requires some minor changes to the contents of the security token. Other than these changes, the metasystem remains precisely the same (including the security token format, message flow, etc.). Table 1 shows the computational load imposed by the scheme on each system party, where E_f denotes a modular exponentiation with respect to a fixed base, M denotes a modular multiplication, and E_v denotes a modular exponentiation to a variable base. Addition and comparison operations have been neglected in these assessments of computational load, because their complexity is much less than that of the exponentiation and multiplication operations.

Table 1: Computational load on system parties

Party	Computational load
IdP	$1E_f$
CEUA	$1E_f + 1M$
RP	$1E_f + 1E_v + 1M$

From Table 1 we conclude that the scheme imposes a manageable computational load on the involved parties, given that modular exponentiations can be performed in milliseconds on modern processors.

The shared parameters p, q and g can be changed frequently if required, and the task of deploying these shared parameters can be achieved using

one of a number of simple methods, e.g. by publishing these parameters on the IdP website. The proposed scheme has the advantage that it does not require any additional key management.

6 Other possible solutions

We now consider three other possible ways in which the two highlighted privacy issues might be addressed. We presented one possible approach in section 4, and we now briefly describe and analyse some other possibilities. Two of these approaches can be combined with the scheme described in section 4, whereas the other is an alternative.

6.1 Using Symmetric Proof-keys

One possible way of addressing the reliance on a single layer of authentication involves use of a long-term secret shared by the IdP and a user. Such a secret could be exploited to provide a second layer of user authentication in a variety of ways; in this section and in section 6.2 we consider two approaches of this type. Both involve making use of the CardSpace proof-key.

The first approach requires a slight modification to the *symmetric proof-key* service outlined in section 2.2. The IdP and the user must first establish a long-term shared secret (k_1), e.g. at the time of user registration with the IdP. This secret could, for example, be generated by the IdP and stored on the user machine or a portable user token (e.g. a USB memory stick or smart card). In order to change this secret (or be re-issued with it), the user would need to conduct a secure exchange with the RP.

In CardSpace, if the *symmetric proof-key* service is used, the IdP includes an asymmetrically encrypted version of a short-term secret (k_2) in the security token, using the RP's public key. We propose that the IdP instead includes an asymmetrically encrypted version of the value $f = h(k_1||k_2)$, where k_2 is a short term secret, h is a cryptographic hash function, and $||$ denotes concatenation of bit strings. The encryption uses the RP's public key, just as in the 'standard' scheme, and the IdP sends the short term secret k_2 to the user, again as in the standard scheme.

The CEUA can now re-compute the value f (using the user stored value of k_1 and the received value of k_2) to prove its rightful possession of the security token. Since this value is computed using the long-term secret k_1, this process adds an additional layer of authentication. That is, if the first layer of authentication between the IdP and the user is broken (e.g. if the password is compromised), the attacker will still need to know the long-term secret in order to access the services offered by the RP. This scheme can readily be integrated into CardSpace.

6.2 Using Asymmetric Proof-keys

A second way of using a long-term secret shared by the user and an IdP to provide a second layer of user authentication involves making a small modification to the *asymmetric proof-key* service, outlined in section 2.2. We present the scheme in the context of a discrete logarithm based asymmetric cryptosystem, although variants for other types of asymmetric cryptosystem may well be possible.

Suppose the user and IdP have agreed on the use of a finite cyclic group G of (large) order q, and a generator g of G, where finding discrete logarithms for elements of G with respect to the base g is computationally infeasible. Suppose also that the user and IdP share a secret integer k (where $0 < k < q$). Then, when requesting a token from the IdP, the user generates a random integer x (where $0 < x < q$), and sends g^x to the IdP instead of sending its public key, as would normally be the case for use of an asymmetric proof-key. The user computes and retains (xk mod q) as its private proof-key. The IdP then computes $g^{xk} (= (g^x)^k)$, and includes this in the token as the asymmetric proof-key public key.

The remainder of the operation of the scheme is then identical to that for the 'standard' asymmetric proof-key scheme. That is, the CEUA possesses a private key corresponding to the public key in the token. The CEUA can then use this private key to prove ownership of the token.

6.3 Modifying Claim Requests

Another way of addressing the problem of untrustworthy RPs avoids the need for the RP to request sensitive information in the security token provided by the IdP. By doing so, the security token then becomes much less privacy sensitive, and hence providing such a token to an untrustworthy RP is no longer a major issue. That is, instead of seeking to avoid providing tokens to untrustworthy RPs, we attempt to minimise the privacy sensitivity of tokens.

Such an approach requires the RP to know the user information in advance (as is the case for the solution discussed in section 4). We exploit this knowledge by modifying the form of the request provided by the RP. That is, instead of asking for user personal information, the RP asks for the IdP to confirm that specific statements about the user's personal information are correct.

The IdP checks the information provided in the request and, if it is all correct, the IdP responds with a token asserting this fact. Such a token is clearly relatively non-sensitive.

Of course, this does mean that user information might be divulged to someone impersonating a user, since the RP submits its request for a token (which now contains potentially sensitive user information) before the user

has been authenticated. It would appear difficult to address this issue without modifying the framework — for example, if the RP knew which IdP was to be used to provide the token, then it could encrypt its request so that only the specified IdP could read it.

6.4 Combining Solutions

To conclude, we briefly consider how the various solutions we have proposed might be combined. Firstly, it would seem clear that the solutions proposed in 6.1 and 6.2 could readily be combined with the SIT-based solution proposed in section 4. Indeed, if the third modification to the SIT-based scheme proposed in section 5.1.4 was adopted, using one of the solutions in 6.1 and 6.2 would be particularly valuable, as otherwise there would be no second layer of user authentication.

By contrast, the idea briefly discussed in 6.3, i.e. where the RP requests confirmation of specified attributes instead of asking for the attribute values, should be seen as an alternative to the SIT-based scheme in section 4, rather than something that complements it. Of course, the scheme in section 6.3 could easily be combined with the schemes in section 6.1 and 6.2, since they address completely different parts of the framework.

7 Concluding remarks

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

We have proposed a modification to address these two security limitations. The proposal involves applying Secured from Identity Theft (SIT) attributes, based on Schnorr's zero-knowledge protocol, to CardSpace. The scheme may be vulnerable to guessing attacks; however, we have also proposed a variety of measures to mitigate the risk of such attacks. Moreover, we also described three possible means by which the CEUA might retrieve the claim values.

The proposed solution can readily be integrated into the current CardSpace, 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, and the involved parties need only perform a small number of inexpensive computations.

Finally, we discussed other possible ways of addressing the two identified security limitations. Two of these approaches are based on the use of the *proof-key* services offered by CardSpace.

References

[1] W. A. Alrodhan and C. J. Mitchell. Addressing privacy issues in CardSpace. In *Proceedings of the Third International Symposium on Information Assurance and Security, IAS 2007*, pages 285–291. IEEE Computer Society, 2007.

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. RFC 4033, Internet Engineering Task Force, March 2005.

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

[4] CA/Browser forum. Guidelines for extended validation certificates — version 1.0 - draft 11, October 2006.

[5] K. Cameron. The laws of identity, May 2005. Microsoft Corporation.

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

[7] 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.

[8] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, NY*, pages 210–217. ACM, 1987.

[9] 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.

[10] 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.

[11] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[12] Microsoft Corporation. Microsoft's vision for an identity metasystem, May 2005.

[13] Microsoft Corporation. A technical reference for InfoCard v1.0 in windows, August 2005.

[14] Microsoft Corporation and Ping Identity Corporation. A guide to integrating with InfoCard v1.0, August 2005.

[15] 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.

[16] 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.

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