Usability and Privacy in Identity Management Architectures

Audun Jøsang **Mohammed AlZomai** **Suriadi Suriadi**

Queensland University of Technology
P.O. Box 2434, Brisbane Qld 4001, Australia
Email: a.josang@qut.edu.au and {alzomaim,s.suriadi}@isi.qut.edu.au

Abstract

Digital identities represent who we are when engaging in online activities and transactions. The rapid growth in the number of online services leads to in an increasing number of different identities that each user needs to manage. As a result, many people feel overloaded with identities and suffer from password fatigue. This is a serious problem and makes people unable properly control and protect their digital identities against identity theft. This paper discusses the usability and privacy in online identity management solutions, and proposed a general approach for making users better able to control and manage their digital identities, as well as for creating more secure identity management solutions. More specifically, we propose a user-centric approach based on hardware and software technology on the user-side with the aim of assisting users when accessing online services.

Keywords: Identity management, user-centric, security, usability, privacy, federation

1 Introduction

An identity is a *representation* of an entity in a specific domain. An identity is normally only associated with a single entity. Shared identities may exist in some sense, for example a family identity that corresponds to several people in a family unit. However, as far as the service provider is concerned, it is dealing with one real world entity (the family). An entity may have several identities within a given domain. For example, a person may be both a parent and a teacher in a school, and thereby have two identities in the school system. The rules for registering identities within a domain determines whether multiple identities for one entity are permitted. Even if forbidden, multiple identities for the same entity may still occur in the system, e.g. in error or because of fraud. An entity can have different identities in different domains. For example, a person may have one identity as a bank customer and another identity as a company employee.

An identity consists of a set of characteristics and attributes, which are called identifiers when used for identification purposes. These characteristics can have various properties, such as being transient or permanent, self-selected or issued by an authority, suitable for human interpretation or only by computers. The possible characteristics of an identity may differ, depending on the type of real world entity being identified. For example, gender applies to people, but not to organisations; a national company registration number applies to a company, but not to a person.

It should be noted that the separation between identity and identifier is blurred in common language. The term "identity" is often used in the sense of "identifier", especially when an identity is recognised by a single unique identifier within a given context. For clarity, the terms "identity" and "identifier" will be used with their separate specific meanings throughout this paper.

A digital identity can be defined as the identity resulting from the digital codification of characteristics and attributes in a way that is suitable for processing by computer systems.

Identity management in computer networks is commonly described as the combination of technologies and practices for representing and recognising entities as digital identities.

A problem with many identity management systems is that they are designed to be cost effective and scalable from the perspective of the service provider (denoted SP hereafter), which sometimes creates inconvenience and poor usability from the consumers' perspective.

In addition to being SP-centric, traditional identity management systems have largely ignored that it is often equally important for consumers to be able to authenticate SPs, as it is for SPs to authenticate consumers.

There is a clear asymmetry between user identity management and SP identity management. In the case of online service provision through the Web, user authentication typically takes place on the application layer, whereas SP authentication takes place on the transport layer through the SSL [1] protocol.

This paper focuses on the usability and privacy aspects of identity management solutions. We will show that some identity models naturally provide better usability and privacy protection, whereas others inherently will expose personal information to third parties that are unrelated to the user-SP relationship. We describe a user-centric approach to identity management, which in our view has superior usability and privacy protection characteristics compared with other models. Our approach was previously described in [1].

[1] Secure Sockets Layer, which is also known as the Transport Layer Security (TLS) in recent Internet standards.

2 Principles of Security Usability

Usability of security is an extremely important, but still poorly understood, element of IT security. An early experimental study is described in [2]. Below we will define some formal security usability principles.

Direct user involvement in a security service is often required, and a distinction can be made between two types of involvement.

- A *security action* is when users are required to produce information and security tokens, or to trigger some security relevant mechanism. For example, typing and submitting a password is a security action.
- A *security conclusion* is when users observe and assess some security relevant evidence in order to derive the security state of systems. For example, observing a closed padlock on a browser, and concluding that the communication is protected by SSL is a security conclusion.

Usability principles related to security actions and security conclusions are described below.

1. Security Action Usability Principles
   (a) The users must understand which security actions are required of them.
   (b) The users must have sufficient knowledge and the practical ability to make the correct security action.
   (c) The mental and physical load of a security action must be tolerable.
   (d) The mental and physical load of making repeated security actions for any practical number of transactions must be tolerable.
2. Security Conclusion Usability Principles
   (a) The user must understand the security conclusion that is required for making an informed decision. This means that users must understand what is required of them to support a secure transaction.
   (b) The system must provide the user with sufficient information for deriving the security conclusion. This means that it must be logically possible to derive the security conclusion from the information provided.
   (c) The mental load of deriving the security conclusion must be tolerable.
   (d) The mental load of deriving security conclusions for any practical number of service access instances must be tolerable.

Although SSL uses strong cryptography, it is for example unable to protect against phishing attacks. This fraudulent practice is perpetrated by attackers posing, for example, as online banks and sending out spam email to people asking them to log on to false, but genuine looking Web sites, which allows the attackers to "phish" identifiers and passwords from unsuspecting users. The problem is not to due to weak authentication mechanisms, but is due to poor usability of the overall Web security solution, of which SSL is only a small part [3]. By comparing the security solution with the security usability principles described above, it can easily be seen why the security fails.

The closed padlock does for example not give sufficient evidence to logically derive the security conclusion of knowing the identity of the SP (violation of security usability principle 2b).

If the user chooses to view the server certificate, the mental load of analysing its content is intolerable (violation of security usability principle 2c).

Depending on the user, it is also doubtful whether many users actually understand what security conclusion they are supposed to make (violation of security usability principle 2a).

This example shows that improved usability, not strengthened cryptography, is needed in order to strengthen users' ability to authenticate SPs in Web interactions.

3 Principles of Online Privacy Protection

Identity management plays a key role for privacy protection, because by definition a digital identity consists of personal information.

Public surveys indicate that privacy is the major concern for people using the Internet [4]. Privacy related complaints that are made to the US Federal Trade Commission include complaints about unsolicited email, identity theft, harassing phone calls, and selling of data to third parties.

A wide array of privacy protection instruments can be used, but none of them will provide adequate protection in isolation.

Legislation of privacy rights has been the main instrument in Europe, but has received much weaker support elsewhere in the world [5].

One attempt to address privacy concerns based on technology is the W3C's Platform for Privacy Preferences (P3P) Project [6]. P3P enables Web sites to express their privacy practices in a standardised, XML-based, format that can be automatically interpreted by user agents such as a Web browser. The aim is that discrepancies between a site's practices and the user's preferences can be automatically flagged. P3P is unable to guarantee or enforce the privacy claims made by Web sites, and has in general received little acceptance in the marketplace.

While it seems that users can not rely on legislation, nor on specific privacy protection technologies such as P3P, users must always practice caution when exposing their personal information. We define the following fundamental user-centric privacy protection principle:

- Exposure of personal information must be minimised.

With relation to identity management solutions, this means that as few parties as possible should be involved in the management of identities used for online service access. The various identity management models described below will be judged against this principle.

4 Existing User Identity Management Models

We will briefly go through existing identity models for managing user identities. Each model will be illustrated from the perspective of usability and scalability. This is done by showing how a user can request services from

different SPs in each model. The diagrams and explanations represent a very high level of abstraction, meaning that the many details of each communication protocol and message are omitted. The diagrams are still sufficiently detailed to show the important aspects of each model from the user perspective.

The diagrams focus in the issuance and management of identifiers and authentication tokens, and explicitly indicate who the identity providers (IdP) and the authentication token providers are. The diagram also shows where the actual authentication takes place.

The various identity management models described below will also be judged against the security usability principles of Sec.2 and the privacy principle of Sec.3.

4.1 The Silo Model

In the silo model[2], the SP manages the name space and authentication tokens for all its users, and therefore takes the role of IdP. The SP also authenticates users based on their identifier-token pairs during service access. Users can be allowed to defined their own identifiers, as long as they are unique within the name space. Users can also define their authentication tokens such as setting the password. This model is illustrated in Fig.1.

The symbol explanations provided by the legend in Fig.1 also apply to subsequent figures whenever relevant.

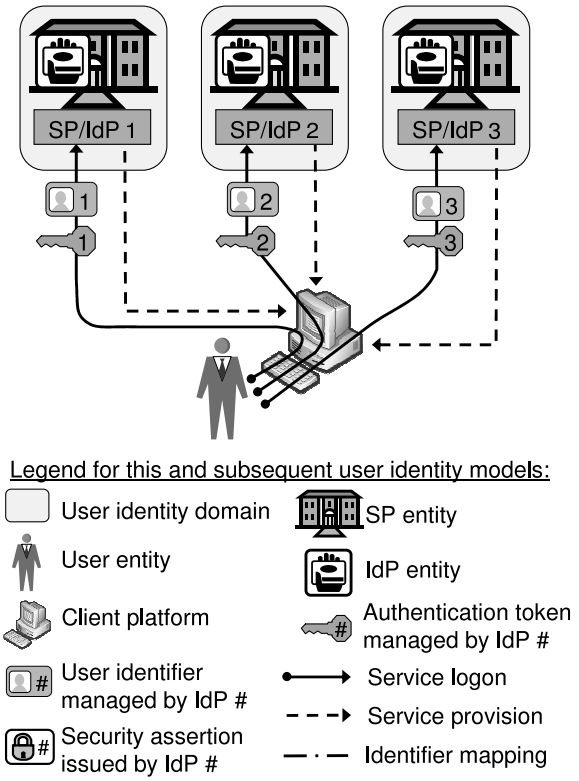

Figure 1: Silo user identity domains

The advantage of this model is that it is simple to deploy for the SPs, and that it exposes personal information only to the SP.

The disadvantage is that it causes identity overload and password fatigue for users when they access services from many different SPs. This violates security usability principle 1d and is illustrated by the multiple login requests originating from the user in Fig.1.

Users routinely forget passwords to infrequently used SPs. Forgotten passwords or fear of forgetting them create a significant barrier to usage, resulting in the services not reaching their full potential. For important sensitive services, where password recovery must be highly secure, forgotten passwords can also significantly increase the cost of service provision.

4.2 The Common Identity Domain Model

In the common identity domain model, a central authority acts as IdP and manages identifiers and authentication tokens, but does not authenticate users during service access. All SPs will identify a specific user based on the same unique identifier from the common name space. The associated authentication token will normally be a public-key certificate. This is illustrated in Fig.2.

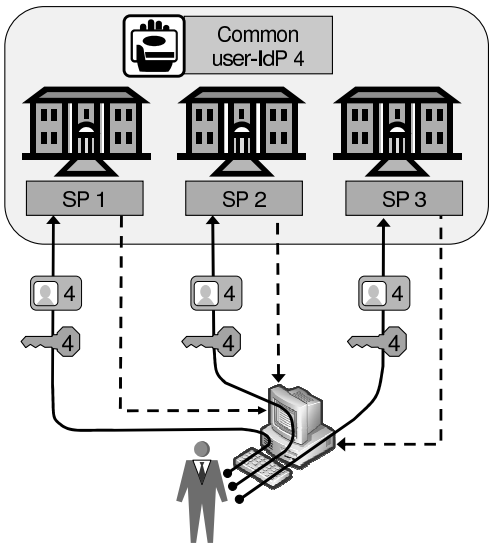

Figure 2: Common user identity domain

The advantage of this model is that it is simple to manage from both the SP and the user perspective. The user only needs to manage a single identifier-token pair.

The main disadvantage is that it is practically and politically impossible to define and manage a stable unique name space for all users worldwide. Email addresses represent an example of a global name space of identifiers. Although email addresses are unique, they are not stable, and many people will have several email addresses. Other nationwide unique name spaces such as e.g. social security numbers are also unsuitable for privacy reasons.

Although users do not need to expose personal information to other parties than the SP, the common unique identifier allows different SPs to match personal information about the same user, which represents a privacy threat.

Practical implementations of this model would need too be based on users having public-key certificates, so that a globally accepted PKI would be needed for managing these certificates.

[2]Called the *"isolated user identity model"* in [1]

4.3 The Centralised SSO Identity Model

As the name indicates, the centralised SSO is used to provide SSO (Single-Sign-On). A single authority manages the name space, issues authentication tokens, and authenticates users during service access. The central authority then sends a security assertion to the SP, either directly as illustrated in Fig.3, or indirectly via the client machine.

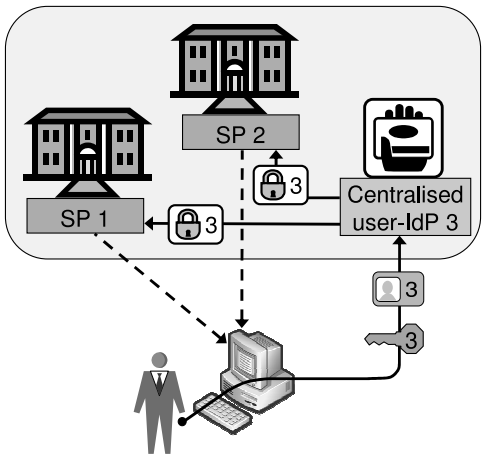

Figure 3: Centralised user identity domain

The advantage of this model is that it provides good usability through SSO. It is well suited for closed networks where multiple SPs are managed by the same organisation, such as in universities, governments and large companies. In closed networks the IdP and the SPs are assumed to be governed by the same authority under a single policy. Authentication networks based on Kerberos and Active Directory are good examples of how organisations can efficiently implement SSO within their corporate networks.

The disadvantage is that it is not suitable for implementation in open environments where SPs are not governed by a common policy and authority. In fact, it is very unlikely that SPs will accept a single IdP to manage identities and do user authenticate on their behalf. It would violate the user-centric privacy protection principle of minimising exposure of personal information. Ideally the centralised IdP should not have access to personal information.

As an example of a centralised SSO for the open global Internet, the Passport authentication service was introduced by Microsoft in 1999. The idea was that users would get a single, convenient method for identifying themselves across different Web sites, and thereby stimulate convenient e-commerce transactions. Using Passport, users would entrust Microsoft to centrally hold their personal information - such as credit card numbers - and make it available to e-commerce websites whenever needed.

However, the market reaction was negative, and Microsoft acknowledged that *"... no single organization, not even one as big as Microsoft, could act as the sole identity provider for everything on the Internet."* [7]. The Passport service was renamed to Windows Live ID in 2006, and currently operates as authentication server for online services controlled by Microsoft such as Hotmail.

From a privacy perspective, the Passport service gave Microsoft significant power to abuse the personal information it collects, and clearly violated the user-centric privacy principle.

4.4 The Centralised Model with Browser Support

In response to the failed attempt to get the Passport service accepted by the whole Internet, and in order to leverage the existing Windows Live ID service (formerly known as Passport), Microsoft is planning to introduce a software and network architecture called InfoCard, where multiple independent IdPs can provide authentication services for e-commerce, and where Windows Live ID simply will be one of multiple such IdPs. The architecture is based on a software module called CardSpace which will be integrated with Internet Explorer 7, planned for release in 2007.

CardSpace [7] provides a storage repository for different identities called InfoCards, that a user might want to use in different situations. For example, a user might want to use different identities when contributing to online blogs, and when purchasing air tickets online. The InfoCards stored in CardSpace on the computer will not contain sensitive information, such as login passwords or credit card information. Each InfoCard points to a centralised IdP where the sensitive information is stored. The IdP will be directly involved in the communication each time a user accesses an online service that requires identity. For example, in the case of accessing an online blog, the IdP can provide a security assertion containing an anonymous login identifier and a password, and in case of purchasing air tickets, the IdP can provide a security assertion containing real name and credit card information. The security assertions are first sent from the IdP to the CardSpace module, and then forwarded to the SPs. There is thus no direct communication between IdP and SP. This architecture is illustrated in Fig.4. Other browser vendors can implement software modules similar to CardSpace in order to support this architecture, and in that sense the architecture itself is not proprietary.

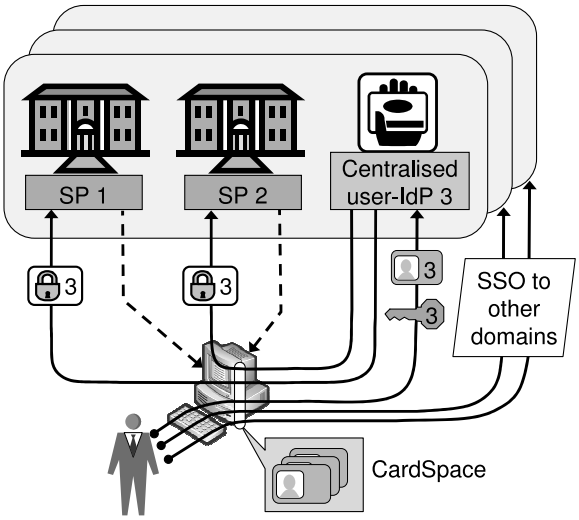

Figure 4: Multiple centralised identity domains

The advantage of having browser support for the centralised model is that it can be simpler for users to access SPs that require identity. By supporting multiple IdPs and multiple centralised domains, the problem of having a single monopoly IdP is diminished.

The main disadvantage of that this is still a centralised model, and that it is unlikely that the market will accept this model any more than they accepted the Passport

model. The only difference is that there are multiple IdPs instead of a single IdP. According to Kim Cameron, identity model architect at Microsoft, *"... online SPs did not see what Passport had to do between them and their customers"* [8]. However, the same applies to any third party IdP. Online SPs will resist an architecture that lets a third party IdP authenticate their users and manage the users' identities unless there is a very close relationship between the SPs and the IdP. From a privacy perspective, it is unacceptable to entrust third parties with personal information when they have no direct involvement in the relationship between the user and the SP. The only realistic usage of this model is to allow closely related SPs, such as when belonging to the same organisation, to outsource the identity management to a common IdP.

Another problem with having multiple centralised domains is that it does not solve the scalability problem for the users. By assuming that multiple centralised identity domains will exist, a user will most likely access SPs from different domains, and will therefore have to authenticate with multiple IdPs. The problem of password fatigue will thus remain, illustrated by the multiple login requests originating from the user to different domains in Fig.4, and this violates security usability principle 1d. This is thus a model that can provide SSO within one identity domain, but not across multiple domains.

A fundamental problem with introducing software modules for identity management such as CardSpace on the client platform is that its security relies on the inherent security of the platform. It is generally accepted that standard computing platforms such as Windows and Linux are relatively simple to compromise, and therefore unsuitable for sensitive transactions. By storing InfoCards with pointers to IdPs on the platform, the potential damage caused by a compromised platform is amplified dramatically. An attacker can steal authentication tokens for accessing IdPs in the same way as passwords are stolen today, e.g. through phishing or Trojans. By also stealing the InfoCards from the client platform, the attacker is able to use those InfoCards and access the users' services without the users' knowledge. This is done by tricking the IdP into sending authentication assertions to SPs. In this way attackers can thus misuse identities without actually stealing them.

Microsoft's motivation for trying to revive the centralised model of Passport through browser support is clear. Acting as a transaction intermediary can provide valuable information which be leveraged to allow new business models. It will for example allow the IdP to collect information about where actual transactions take place on the Internet. By anonymising this information, it will not necessarily represent a threat against user privacy.

4.5 The Federated SSO Identity Model

Federated identity models are based on groups of SPs that enter into a mutual security and authentication agreement in order to allow user SSO to their services. In the terminology of the Liberty Alliance, these groups are called *circles of trust* [9].

Identity federation can thus be defined as a set of agreements, standards and technologies that enable SPs to recognise user identities and entitlements from other SPs.

As in the silo identity model, each SP manages the name space of all its users. Various silo identity domains are then linked together to form a federated domain. In practice, the different domain specific identifiers of each user are mapped, as illustrated in Fig.5. This makes it possible for different SPs to refer to the same user with different identifiers, For authentication purposes, an SP will accept a *security assertion* [10] from another SP claiming that a given user has already been authenticated.

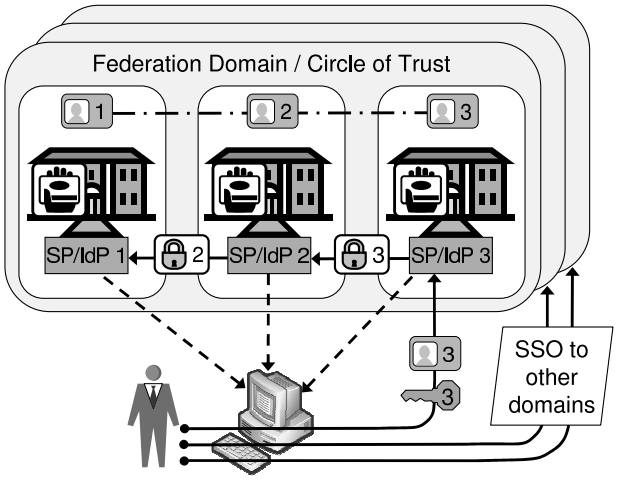

Figure 5: Federated user identity domains

The main advantage of the federated identity model is that it provides SSO in open environments. The federated model is compatible with, and can be retrofitted to the traditional silo model. SPs can thus keep existing name spaces and authentication systems.

The main disadvantage is that it creates legal and technical complexity. For example, SPs are technically speaking not able to distinguish between a security assertion that reflects a genuine user service request, or one that represents a SP masquerading as a user. SPs must therefore trust security assertions from each other. In addition, the mapping between identifiers will technically speaking allow SPs to correlate information about the same user, which could represent a privacy threat. Users must therefore trust SPs to protect their privacy in this respect.

From a privacy perspective, a federated model can be both an advantage and a disadvantage. Different SPs within the same federation domain are technically able to match personal information of the same user because of the mapping between identifiers. The privacy protection thus depends on the privacy policy and the adherence to the policy, which can pose a threat. On the other hand, a user's identity within a specific SP's silo domain can be anonymous, and only the "home" SP needs to know the real world identity of the user. This can provide additional privacy protection.

Similarly to having multiple centralised identity domains as described in Sec.4.4, federated identity domains can not solve the scalability problem for the users. By assuming that multiple federated identity domains will exist, a user will most likely access SPs from different domains, and will therefore have to authenticate with the respective IdPs from each domain. The problem of password fatigue will thus remain, illustrated by the multiple login requests originating from the user in Fig.4, and this violates security usability principle 1d. This is thus a model that can provide SSO within each federated identity domain, but not across multiple federated domains.

5 User-Centric User Identity Management

In this section we describe our approach to management of user identities. It solves the scalability problem, and has the potential of providing a universal SSO solution while still being compatible with the other models described in Sec.4 above. The user-centric approach also provides stronger security than traditional solutions are able to provide.

This is achieved by having a separate hardware device called PAD (Personal Authentication Device) that can take active part in security of transactions, and that can store identity information and authentication tokens.

It is important to distinguish between entity (user) authentication, and message authentication. Entity authentication is when a subject entity proves the possession of an authentication token when accessing a service which often is defined in terms of a session. This indicates that the entity was present at some network access point at some moment in time. However, it does not prove that the entity is continuously present at the same access point during the whole session, nor that every message that enters the network through that access point originates from the user. It is for example possible that the human user walks away from the client terminal during an active session, and that another person takes his place and sends access requests from the session. It is also possible that a Trojan program hiding on the client platform initiates activities from the same session without the user even knowing.

In order to prevent hijacking of the session, i.e. session activities that do not originating from the legitimate user, message authentication is needed. This means that simple user authentication is insufficient, and that the user must authenticate every security critical message sent from the client. This can be required for sensitive transactions such as online banking, and many banks have already introduced solutions for message authentication, as described below.

5.1 SMS Authorisation Codes

Some banks issue special hardware tokens that can generate one-time authorisation codes, whereas other banks rely on out-of-band communication to the customer's mobile phone. We describe the latter scenario below.

An interesting solution which has been implemented by the National Australia Bank, is based on authorising bank transactions using SMS messages sent to the user's mobile phone. Although the user has been authenticated and is already logged in, this allows authentication of the transaction request itself.

SMS messages sent from the bank to the user's mobile phone pass through the cellular network, and is independent of the Internet. The user can manually transfer data from the mobile phone to the client terminal. By verifying the correct transfer of data from the mobile phone to the client terminal, the bank can conclude that the user received the data through the cellular network, read it and submitted it through the Internet. This is then interpreted as a genuine intent to submit the transaction. The security of this scheme is based on the assumption that it is difficult for an attacker to steal the user's personal mobile phone and to attack the cellular network. The scenario is illustrated in Fig.6.

The SMS authorisation code is computed as a function of the origin and destination accounts, as well as the amount. It typically consists of 8 digits which is the same length as a normal telephone number, and can therefore be copied manually from the mobile phone to the client terminal without too much effort. A typical SMS message as used in the scheme of National Australia Bank is illustrated in Fig.7.

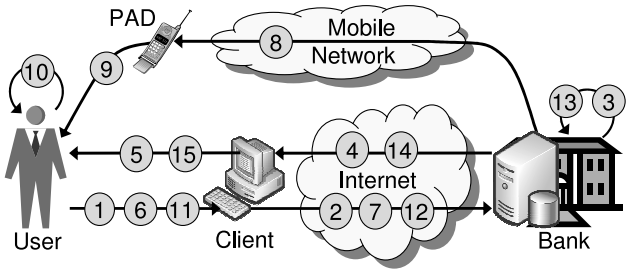

Figure 6: Authorising bank transactions via SMS

Table 1: Messages in SMS bank security protocol

Msg ♯	Message description
1	Produce Login Id and authentication token
2	Transmit Login Id and authentication token
3	Verify Login Id and authentication token
4	Transmit service options
5	Present service options
6	Transaction request
7	Transmit transaction request
8	SMS message with authorisation code
9	Read SMS message
10	Verify amount and bank account number
11	Copy authorisation code
12	Transmit authorisation code
13	Verify authorisation code
14	Transmit transaction confirmation
15	Present transaction confirmation

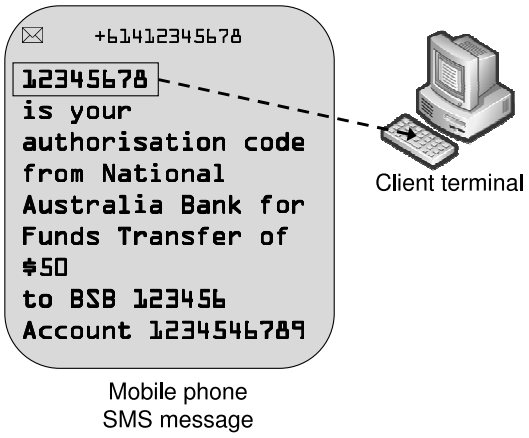

Figure 7: Example SMS message with authorisation code

Assuming that an attacker changes the amount and/or the destination account number, e.g. by a Trojan program on the client terminal, the modified amount and account number will appear in the SMS message. It is assumed that the correctness of the amount and of the destination account number is verified by the user when copying the

authorisation code from the SMS message. However, this can be quite tedious and could therefore violate security usability principle 2c and 2d. If a user victim fails to notice that the bank account number in the SMS message is not the same as the intended account number, and submits the authorisation code through the client terminal, the attack will succeed. Despite being the victim of an attack, the liability could be put on the user because he accepted the SMS message. The usability of SMS authorisation needs to be investigated in order to determine whether the user can be made liable for this type of attacks.

Assuming that the user is able verify the correctness of the amount and of the bank account number in SMS messages consistently and reliably, this scheme is secure against attacks on the client terminal, and is in fact independent of the security of the client terminal. This would represent a considerable security improvement.

This scheme assumes that the mobile terminal can be trusted, i.e. that no attacker is able to take over the control of mobile terminals, in contrast to standard desktop client terminals. If it were possible to take over the control of the mobile terminal, an attacker could change the SMS message, and present the expected amount and the bank account number, so that the SMS message that the user reads is not the same as the SMS message that the bank sent.

The scheme also depends on the security of the mobile phone networks, and it assumes that no attacker is able to modify SMS messages sent to the user while in transit through the mobile network. Even if interception and cryptanalysis of SMS messages sent over the air were possible, it requires that the the attacker is physically present in the same base station coverage area, and this excludes attacks from anywhere in the world.

5.2 User-Centric Silo Model

The user-centric model can be combined with any of the traditional identity management models described in Sec.4. Fig.8 illustrates the user-centric approach combined with silo user identity domains. A user centric approach combined with e.g. federated identity domains would also be meaningful.

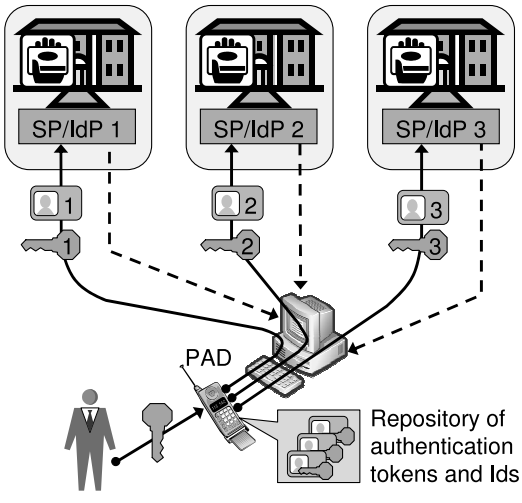

Figure 8: User-centric identity model with silo domains

The user only needs to login to the PAD once. The PAD is a single-user device, so no identifier is needed. Once logged in to the PAD, the user can access any service, and the PAD does the login to each SP on the user's behalf. This can thus provide a universal SSO solution.

When combined with the silo model described in Sec.4.1, the user-centric approach provides the same privacy protection as the silo model because exposure of personal information is limited to each SP with which the user transacts. The user-centric approach can also be combined with any other user identity model described above, and will then inherit the privacy properties of these models.

From a usability point of view, the user-centric approach solves the scalability problem. All the identifiers and authentication tokens can be stored in the PAD, and the PAD can also be configured to be directly integrated in authentication protocols. Similarly to the SMS-based security solution described in Sec.5.1, the PAD can make the security independent of the security of the client platform.

The term *Personal Authentication Device* has been in use within the context of computer security at least since 1985 (Wong, et al., 1985). While the details of the operations and limitations of the devices have varied significantly since that time, the key concepts remain the same. A more recent incarnation of the same concept can be found in the form of the Personal Trusted Device defined in the context of the *Personal Transaction Protocol* [11].

For all practical purposes, the PAD can be a mobile phone or a PDA (Personal Digital Assistant). Because the PAD provides a security service, it is crucial that the PAD platform itself is secure.

The fundamental security problem with common computing platforms such as Windows and Linux is that there is no robust isolation of processes. Compromise of one application often leads to the compromise of the whole platform. This architecture, whereby processes can directly access each other is considered a benefit in a general purpose computing platform because of the flexibility it facilitates. For sensitive applications however, this architecture is not appropriate.

It is likely that mobile manufacturers are tempted to increase the flexibility and connectivity of mobile phones [12] in order to allow new business models. Unfortunately this will necessarily lead to the introduction of security vulnerabilities and make such devices unsuitable as PAD. Viruses for mobile phones are for example already common [13]. It must be assumed that the PDA provides strong isolation of processes, in which case it can be called a trusted platform. If the PAD is combined with a mobile phone, the functionality must be restricted in comparison with many mobile phones that are currently on the market.

As indicated in Fig.8, the user needs to authenticate to the PAD before it can authenticate to SPs. The PAD can for example require two-factor authenticate through a combination of PIN and biometrics, or through PIN and a smartcard. Authentication to SPs will then actually be three-factor because it also requires having the PAD itself.

The PAD should support multiple types of authentication tokens, and be integrable in different types of authentication protocols. In its simplest form, the PAD could simply store identifiers and passwords that can be read by the user and copied to login screens in the client platform. More sophisticated solutions can directly involve the PAD in the communication between SP and client. By letting the PAD communicate directly with the SP it is possible to design *multi-channel security protocols*.

6 SP Identity Management Models

In this section we will briefly describe models for managing SP identities. The models will be seen from the point of view of how different users can authenticate the same SP.

6.1 Common SP Identifier Domain

The only widespread model for authenticating SPs on the Internet is by having a common identity domain recognised by all users. The name space of unique identifiers for SPs simply consists of all the Internet domain names. The name space, together with the technology and organisations that manage it, are commonly called the DNS (Domain Name System). The DNS forms a hierarchical tree under the supervision of ICANN (Internet Corporation for Assigned Names and Numbers), where the authority for each sub-domain manages the DNS directly underneath it. Briefly said, the DNS is the IdP in this model.

The authentication tokens consists of the so-called server certificates, which are issued by CAs (Certificate Authorities). The server certificates are also organised in a hierarchical fashion, and the root certificate of each CA is hard-coded and shipped with every Web browser. This multi-hierarchy of server certificates can simply be called the Browser PKI, and represents a common structure for distributing authentication tokens to SPs.

The architecture for authenticating SPs is illustrated in Fig.9.

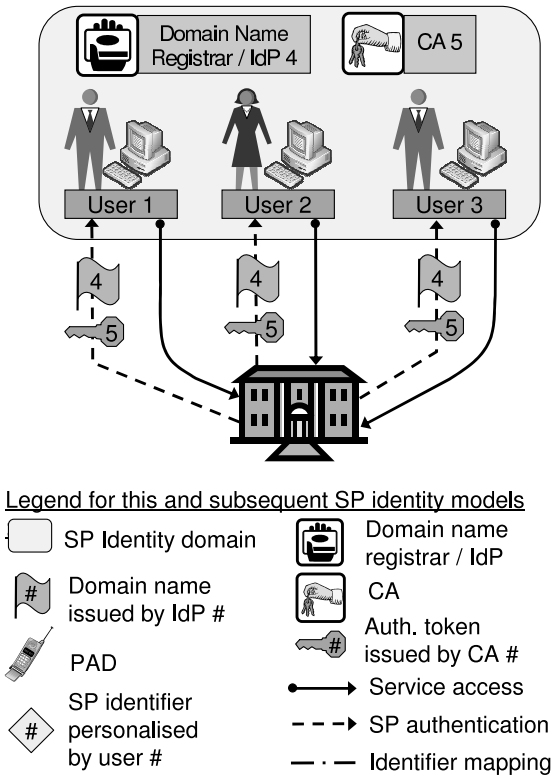

Figure 9: The DNS identity domain and the browser PKI

The U.S. government acquired responsibility for, and authority over the DNS root by virtue of its historical initiative and financial investment in supporting creation of the Internet and the DNS. However, as the Internet has become international in extent, support and operation, the formal legitimacy of the U.S. government's continued authority over the root has come under challenge[14], such as in the World Summit on the Information Society (WSIS)[3]

Although the approach illustrated in Fig.9 is based on strong cryptography, it has been successfully attacked, e.g. in the form of phishing attacks. This is due to the poor usability of the Browser PKI and SSL security in general. The SP identity authenticated by the client terminal is not necessarily the same as the SP identity intended by the user, and there is no simple way for the user to find out which SP the client in fact has authenticated.

6.2 User-Centric Management of SP Identities

Assuming that each user owns a PAD as described in Sec.5.2, the users can create private identifiers for SPs by mapping the global unique identifiers, such as a domain name, to personally chosen identifiers. This identifier can be anything that can be practically recognised, e.g. text, pictures, logos and sound. This is illustrated in Fig.10.

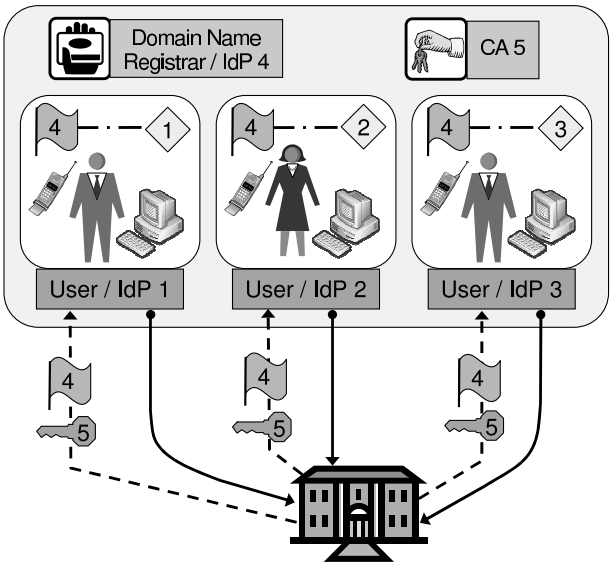

Figure 10: User-centric approach to SP identity management

Fig.10 shows the authentication protocol from a high level of abstraction, and does for example not show the communication between the user and the client terminal. A more detailed scenario is described in Sec.7 below.

The index "4" of the identifier indicates that it has been assigned by the Domain Name Registrar 4, and the index "5" of the authentication token sent with the authentication messages of Fig.10 indicates that it has been issued by CA 5. The authentication tokens are thus ordinary X.509 server certificates. The indexes "1", "2" and "3" of the SP identifiers in the personal domains indicate that these have been assigned by the respective users.

The mapping between the global SP identifier and the personal SP identifier takes place within the user domain. To be practical, this requires the user terminal or PAD to be directly involved in the authentication protocol in some way. There are many ways of achieving this, and each practical solution will depend on the type of device and network connection.

[3] See http://www.itu.int/wsis/.

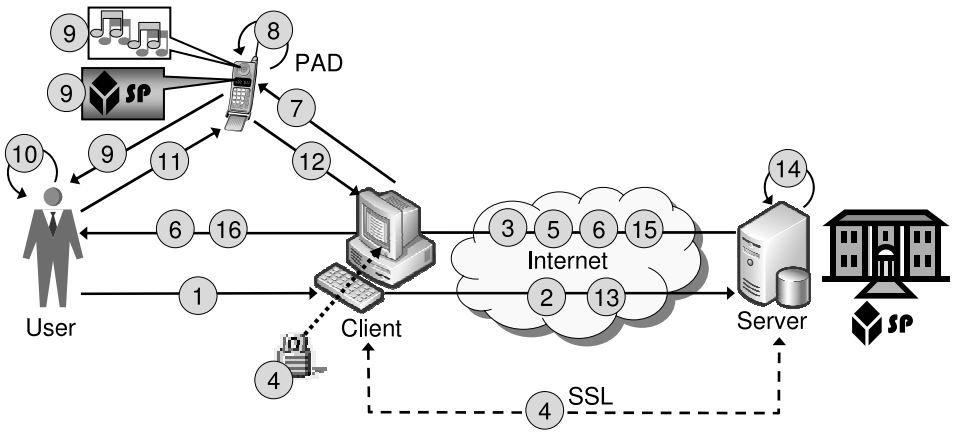

Figure 11: Mutual authentication scenario with PAD

The Mozilla TrustBar [15] is a software implementation of this concept. The TrustBar is a plug-in toolbar for the Mozilla and Firefox browsers, where the user can store images mapped to server certificates. Each time a server certificate is successfully verified, the toolbar will check if a mapping exists, and display the mapped image on the toolbar while the corresponding page is loading.

The mapping between global and personal SP identifiers can also take place in a separate hardware device such as the hardware PAD described in Sec.5.2. As an example, assume that the PAD is embedded in a mobile phone, and that the user accesses a service through the Internet using an ordinary computer terminal. Mutual authentication between the SP and the user can now take place by combining the IP channel with mobile channels. When registering with the service, the user must specify through which channels he wants authentication to take place. Assuming that the user has assigned and mapped a private SP identifier in the form of an image or company logo to the SP identifier in the certificate, the PAD / mobile phone can display the image on the screen when the certificate has been verified by the PAD. Since the user chose the image in the first place, one can assume that the user is able to recognise the same image when authenticating the SP at a later stage.

This model effectively eliminates the phishing attack threat, as illustrated with the following example. Assume that the attacker has purchased a genuine certificate in order for an SSL channel to be established, the SSL padlock will be displayed on the browser window when accessing the attacker's server. Assume that a user responds to a spam email message by clicking on a URL pointing to the attackers server, in the belief that it points to the genuine server. Even if the certificate is correctly verified by the browser or by the PAD, it will not be mapped to anything, and the TrustBar or the PAD will give a warning, and thereby indicate to the user that the Web site is unknown.

7 Combining User and SP Identity Management

In the previous sections we have described how a user-centric approach can be applied to the management of both user identities and SP identities. The strength of the user-centric approach becomes fully evident when both are combined.

From a privacy perspective, the user centric approach reduces to a minimum the number of parties to which personal information is disclosed. From a usability and security perspective, the PAD can be used to personalise server certificates so that they are more easily recognised.

A simple scenario can for example provide mutual authentication based on the user-centric silo identity identity model of Fig.8 and the user-centric common SP identity model of Fig.10. This is illustrated in Fig.11. The messages of Fig.11 are described in Tab.2.

Table 2: Messages in mutual authentication protocol

Msg ♯	Message description
1	Specify service request
2	Transmit service request
3	Transmit server certificate
4	Establish SSL connection
5	Transmit login page
6	Present login page
7	Transmit server certificate to PAD
8	Verify server certificate
9	Play/display personalised certificate info
10	Verify correctness of personalised info
11	Accept SP
12	Transmit identifier and authentication token
13	Transmit identifier and authentication token
14	Verify identifier and authentication token
15	Submit service page
16	Present service page

It is assumed that the user has already authenticated to the PAD, and Msg.(11) *"Accept SP"* of Fig.11 does does not require any password or other authentication token. This is thus a SSO scheme, whereby the PAD automatically transmits the user identifier and authentication token on behalf of the user.

7.1 Potentials of User-Centric Identity Management

By providing identity management assistance to users in the form of a PAD, many authentication scenarios are possible. Fig.12 illustrates all the possible communication channels that can be leveraged to create secure and user friendly solutions.

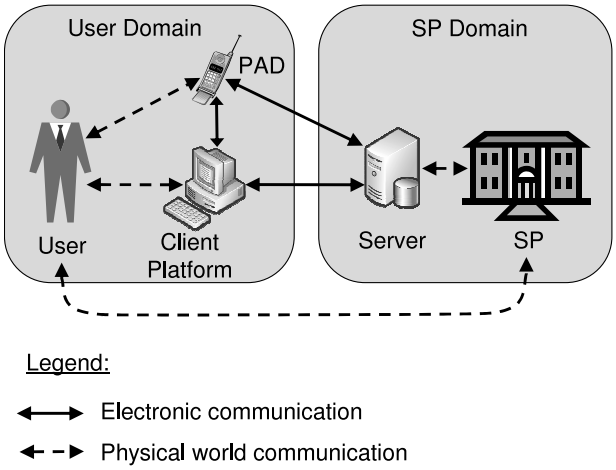

Figure 12: Possible communication channels when using a PAD for identity management

The topology of the graph in Fig.12 allows 4 separate electronic communication paths between the user and the SP. The authentication architecture based on SMS described in Sec.5.1 uses only two of these paths. In addition, each communication link can be based on different protocols and technology. The communication between the PAD and the client terminal can for example take place through a USB cable or via Bluetooth. This approach thus opens up a large number of different architectures, and allows multi-channel security protocols.

When designing security protocols, the computational complexity of tasks to be executed by any of the nodes is an important aspects. That is for example why public-key encryption is never used for encrypting bulk data.

When human users form part of a security protocol, similar considerations need to be made. More specifically, the mental load of the task to be performed by the human user must be taken into account. Even if the load of a single task is tolerable, the load of performing the same task repeatedly must also be considered. The security usability principles described in Sec.2 can be used as a guideline.

8 Conclusion

There will always be a trade-off between different goals when designing identity management solutions, and it is natural that the service and infrastructure providers promote solutions that seem immediately advantageous to them. However these solutions are not necessarily beneficial from the usability and privacy perspective, and this in turn becomes a problem for all parties.

Our study has demonstrated that a user-centric approach to identity management has the greatest potential of providing good usability combined with strong privacy protection. By being compatible with legacy models, this represents a good solution for all parties.

References

[1] A. Jøsang and S. Pope. User-Centric Identity Management. In Andrew Clark., editor, *Proceedings of AusCERT 2005*, Brisbane, Australia, May 2005.

[2] A. Whitten and J.D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, Washington, D.C., August 1999.

[3] A. Jøsang, P.M. Møllerud, and E. Cheung. Web Security: The Emperors New Armour. In *Proceedings of the European Conference on Information Systems (ECIS2001)*, Bled, Slovenia, June 2001.

[4] A. Cavoukian and M. Crompton. Web Seals: A Review of Online Privacy Programs. A Joint Project of The Office of the Information and Privacy Commissioner/Ontario and The Office of the Federal Privacy Commissioner of Australia, http://www.ipc.on.ca/english/pubpres/papers/seals.pdf, Venice, September 2000.

[5] L.A. Bygrave. Privacy Protection in a Global Context - A Comparative Overview. *Scandinavian Studies in Law*, 47:319–348, 2004.

[6] L. Cranor et al. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Recommendation 16 April 2002, http://www.w3.org/TR/P3P/, 2002.

[7] David Chappel. Introducing Windows CardSpace. http://msdn.microsoft.com/library/en-us/dnlong/html/IntroInfoCard.asp, April 2006.

[8] Kim Cameron. Interview with Kim Cameron about the Identity Laws. Webcast available from: http://channel9.msdn.com/, June 2006.

[9] Liberty-Alliance. *Liberty ID-FF Architecture Overview.* Version: 1.2-errata-v1.0. http://www.projectliberty.org/specs/liberty-idff-arch-overview-v1.2.pdf, 2003.

[10] OASIS. *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0,* Committee Draft. Organization for the Advancement of Structured Information Standards, 15 January 2005.

[11] Mobile Electronic Transactions Ltd. *Personal Transaction Protocol Version 1.0,* Draft Specification 01-11-2002. MeT, 2002.

[12] A. Jøsang and G. Sanderud. Security in Mobile Communications: Challenges and Opportunities. In *The Proceedings of the Australasian Information Security Workshop*, Adelaide, February 2003.

[13] Jim Krane. As mobile devices get 'smarter', they become prone to viruses. SiliconValley.com - Mercury News, URL: http://www.siliconvalley.com/mld/ siliconvalley/2833740.htm, 10 March 2002.

[14] National Research Council. *Signposts in Cyberspace - The Domain Name System and Internet Navigation*. The National Academic Press, Washington, D.C., 2005.

[15] Amir Herzberg and Ahmed Gbara. Protecting (even Naïve) Web Users from Spoofing and Phishing Attacks. Technical Report 2004/155, Cryptology ePrint Archive, 2004.