User Centric Identity Management

Audun Jøsang and **Simon Pope**

CRC for Enterprise Distributed Systems Technology (DSTC Pty Ltd)*
The University of Queensland, 4072, Australia
{ajosang, simon.pope}@dstc.edu.au

Abstract

Identity management is traditionally seen from the service providers' point of view, meaning that it is an activity undertaken by the service provider to manage service user identities. Traditional identity management systems are designed to be cost effective and scalable primarily for the service providers, but not necessarily for the users, which often results in poor usability. Users are, for example, often required to memorise multiple passwords for accessing different services. This represents a minor inconvenience if users only access a few online services. However, with the rapid increase in the uptake of online services, the traditional approach to identity management is already having serious negative effects on the user experience. The industry has responded by proposing new identity management models to improve the user experience, but in our view these proposals give little relief to users at the cost of relatively high increase in server system complexity. This paper takes a new look at identity management, and proposes solutions that are designed to be cost effective and scalable from the users' perspective, while at the same time being compatible with traditional identity management systems.

1 Introduction

When making services and resources available through computer networks, there is often a need to know who the users are and to control what services they are entitled to use. In this context, identity management has two main parts, where the first consists of issuing users with credentials and unique identifiers during the initial registration phase, and the second consists of authenticating users and controlling their access to services and resources based on their identifiers and credentials during the service operation phase. A problem with many identity management systems is that they are designed to be cost effective from the perspective of the service providers (SP), which sometimes creates inconvenience and poor usability from the users' perspective.

In addition to being SP centric, traditional identity management systems have largely ignored that it is often equally important for users to be able to identify service providers, as it is for service providers to authenticate users. 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 protocol.

*The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

However, the common scam called *password phishing* illustrates the difficulty of service provider authentication with SSL. The 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 current the SSL security model. Although strong cryptographic mechanisms are being used, it can be difficult for users to know which SP identity has been authenticated. Improved usability, not strengthened cryptography, is needed in order to strengthen users' ability to authenticate service providers in Web interactions.

This paper describes an emerging approach, called user-centric identity management, that focuses on usability and cost effectiveness from the users' point of view, and that is also compatible with traditional identity management models.

2 Identity and Related Concepts

An identity is a representation of an entity in a specific application domain. For example, the registered personal data of a bank customer, and possibly also the customer's physical characteristics as observed by the bank staff, constitute the identity of that customers within the domain of that bank. Identities are usually related to real world entities. Typical real world entities are people or organisations. A simplifying assumption is that a single identity can not be associated with more than one entity. Shared entities may exist, 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) and not with multiple individuals. A person or organisation may have zero or more identities within a given domain. For example, a person may have two identities in a school system because he or she is both a parent and a teacher at the school. The rules for registering identities within a domain determine 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. A person may of course have different identities in different domains. For example, a person may have one identity associated with being customer in a bank and another identity associated with being an employee in a company.

An identity consists of a set of characteristics, which are called identifiers when used for identification purposes. These characteristics may or may not be unique within the identity domain. They 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, a date of birth applies to people, but not to organisations; a national company registration number applies to a company, but not to a person.

The relationship between entities, identities and characteristics/identifiers are shown in Fig.1 below.

The figure illustrates that an entity, such as a person or an organisation, may have multiple identities, and each identity may consist of multiple characteristics that can be unique or non-unique identifiers.

It should be noted that the separation between identity and identifier is blurred in common language usage. The term "identity" often is 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.

An identity domain is a domain where each identity is unique. A name space of unique identifiers in a domain allows a one-to-one relationship between identities and identifiers. Not every identity characteristic can be used as unique identifiers: for example, a date of birth does not uniquely identify an individual person, because two or more people can have the same date of birth. A name space of unique identifiers is usually designed based on specific criteria which, for example, could be that the identifiers

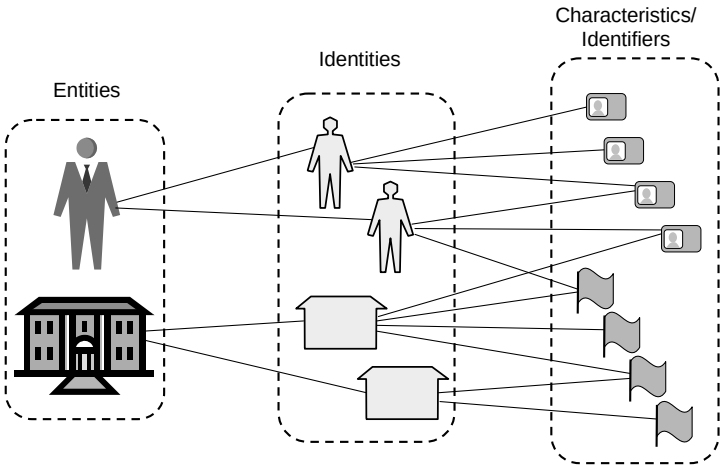

Figure 1: Correspondence between entities, identities and characteristics/identifiers.

must be suitable for memorisation by a human or only readable by a computer, that all identifiers have a fixed length or that they can have flexible length etc. It can be quite challenging to define a good name space, and in general, the larger the domain (i.e the more entities one needs to identify), the more difficult it is to define a suitable name space of unique identifiers. For example, a name space of unique identifiers for all humans seems to be politically and practically impossible to achieve. Name spaces must be carefully designed, because a poor name space design that must be changed at a later stage can result in significant extra costs. For example, when it became clear that the current 32 bit name space of fixed length Internet Protocol addresses in IPv4 would become to small, a new name space with 128 bits was designed for IPv6, with the result that IPv4 and IPv6 addresses are incompatible.

A pseudonym may be used as unique identifier in some systems for privacy reasons in order to provide an anonymous identity [1]. The pseudonym is an identifier where only the party that assigned the pseudonym knows the real world identity behind it. The pseudonyms can be self-assigned, so that the real world identity (e.g. legal persona) behind the pseudonym is only known by the owner, and otherwise is hidden to all other parties. Alternatively, the pseudonym can be defined and escrowed by a trusted third party who knows the real world identity, and who is able to reveal it under special circumstances such as law enforcement.

3 Traditional User Identity Management Models

In order to better understand the merits of the user-centric approach to identity management described in later sections, this section takes a closer look at traditional models and current practice.

3.1 Isolated User Identity Model

The most common identity management model is to let service providers act as both credential provider and identifier provider to their clients. They control the name space for a specific service domain, and allocate identifiers to users. A user gets separate unique identifiers from each service/identifier provider he transacts with. In addition, each user will have separate credentials, such as passwords associated with each of their identifiers. This model, which can be called *isolated user identity management*, is illustrated in Fig.2 below.

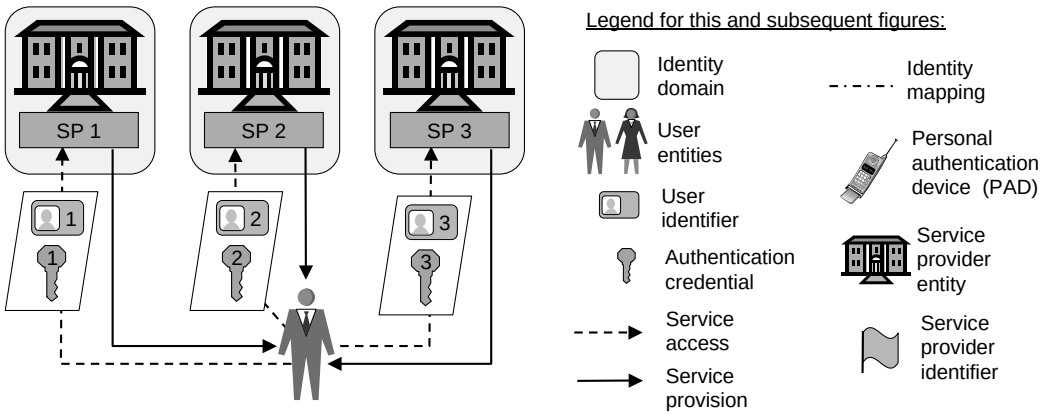

Figure 2: Isolated user identity model.

The identifier and credential indexes in the figure refer to the issuing entity. For example, an identifier and credential with index 1 means that it has been issued by SP 1.

This approach might provide simple identity management for service providers, but is rapidly becoming unmanageable for users. The explosive growth in the number of online services based on this model results in users being overloaded with identifiers and credentials that they need to manage. Users are often required to memorise passwords, which unavoidably leads to users forgetting passwords to infrequently used services. Forgotten passwords, or simply the fear of forgetting, create a significant barrier to usage, resulting in many 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 for the service providers.

3.2 Federated User Identity Model

he *federated identity management model* attempts to address the type of inefficiencies described in Sec.3.1 above. Identity federation can be defined as the set of agreements, standards and technologies that enable a group of service providers to recognise user identifiers and entitlements from other service providers within a federated domain.

In a federated identity domain, agreements are established between SPs so that identities from different SP specific identity domains are recognised across all domains. These agreements include policy and technology standards. A mapping is established between the different identifiers owned by the same client in different domains, that links the associated identities. This results in a single virtual identity domain, as illustrated in Fig.3. When a user is authenticated to a single service provider using one of their identifiers, they are considered to have been identified and authenticated with all the other service providers as well. This happens by passing assertions between service providers.

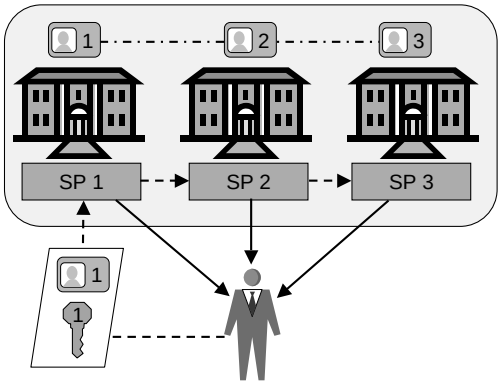

Figure 3: Federated user identity model.

Sending an assertion does not require user credentials, and the acceptance of user access assertions

from one SP to another is based on trust established by adherence to common policies.

The federation of isolated identifier domains gives the client the illusion that there is a single identifier domain. The user can still hold separate identifiers for each service provider. However, he does not necessarily need to know or possess them all. A single identifier and credential is sufficient for him to access all services in the federated domain. This can therefore be used to provide a Singel-Sign-On (SSO) solution similar to that described in Sec.3.3.3. However, a potential problem is that users will still have to manage multiple identities and credentials, even if they are not actively using all of them. Therefore, identity federation makes most sense when the user wants to manage only one set of identifiers and credentials.

Technology standards for identity federation include the OASIS Security Assertion Markup Language (SAML) [8] and the Liberty Alliance framework [6]. Shibboleth [10] is an open source implementation of the federated identity management model. Major vendor are also offering federated identity management solutions.

3.3 Centralised User Identity Models

In centralised user identity models, there exists a single identifier and credentials provider that is used by all service providers, either exclusively, or in addition to other identifier and credentials providers. Centralised identity models can be implemented in a number of different ways. Below we describe the common identifier model, the meta-identifier model, and the single sign-on (SSO) model.

3.3.1 Common User Identity Model

A relatively simple identity management model is to let a separate entity or single authority act as an exclusive user identifier and credentials provider for all service providers. This architecture, which can be called the *common user identity management model*, is illustrated in Fig.4.

In the common user identity model, a user can access all service providers using the same set of identifier and credential. This could, for example, be implemented by having a PKI where a single Certificate Authority (CA), or subordinate or cross certified CAs thereof, issue certificates to all users within the domain. The identifier name space can for example be the set of Internet email addresses that in fact are globally unique. Assuming that all the criteria necessary to operate a PKI are satisfied (which is far from trivial), users only need a single set of identifier and credential to be authenticated by all service providers.

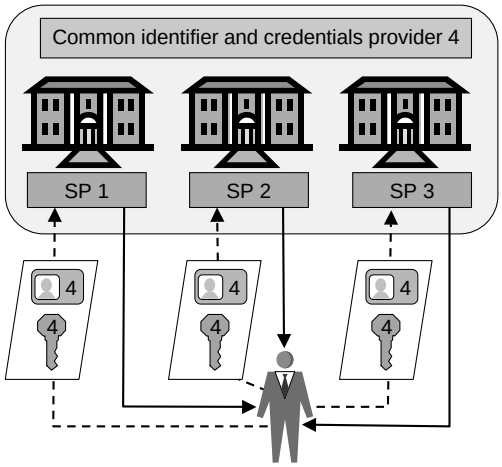

Figure 4: Common user identity model.

On a global scale it would be problematic to use email addresses as unique identifiers. For example, email addresses can be obtained anonymously, people can change email address whenever they like, and the same person can have many email addresses simultaneously, which would be unacceptable for many applications. On a smaller scale, such as within a single organisation where the assignment of email addresses can be controlled, this model could work well.

3.3.2 Meta User Identity Model

Service providers can share certain identity related data on a common, or meta, level. This can be implemented by mapping all service provider specific identifiers to a meta identifier with which for example the credential can be linked. This is illustrated in Fig.5.

The meta identifier approach is commonly implemented by a so-called meta directory, and is a popular approach for integrating legacy identity management systems in large enterprises. In this case, all the services linked to the meta identity domain are usually under the administration of a single organisation or authority.

In theory the meta identity model can also provide an integrated identity management approach for different service providers, but that would require policy alignment and strong trust between the involved parties.

The unique meta identifier is normally hidden from users and only used internally for identity management and service coordinating purposes. From a user perspective, this can be seen as password (or credential) synchronisation across multiple service providers. When the user changes the password with one service provider, it is automatically changed with all the others as well.

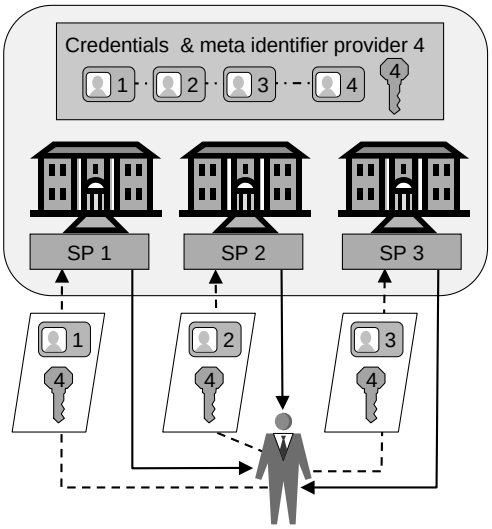

Figure 5: Meta user identity model.

3.3.3 Single-Sign-On Identity Domain

A simple extension of the centralised identity management approaches described in Sec.3.3.1 and Sec.3.3.2 could be to allow a user authenticated by one service provider, to be considered authenticated by other service providers. This is commonly called a Single Sign-On (SSO) solution because the user then only needs to authenticate himself (i.e. sign on) once to access all the services.

There will normally be one party responsible for allocating identifiers, issuing credentials and performing the actual authentication as illustrated in Fig.6.

This SSO scenario is very similar to the federated identifier scenario described in Sec.3.2, except that no mapping of user identifiers would be needed because the same identifier is used by every service provider. Kerberos based authentication solutions, where the Kerberos Authentication Server acts as the centralised identifier and credential provider, are in this category. Microsoft .Net Passport is an example of an SSO implementation for e-commerce, where email addresses are adopted as user identifiers. In the .Net Passport model, credential issuance and authentication are centralised functions under Microsoft's control.

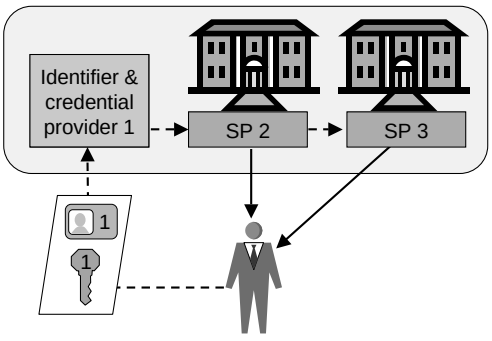

Figure 6: SSO identity model.

4 User Centric User Identity Management

An authentication solution must take into consideration how the identifiers and credentials are to be handled by the user. If the usability is poor, then the authentication itself will be weak because users are unable to handle their credentials adequately. In this regard, it is interesting to notice that service providers usually have automated systems to manage identities and authentication, whereas users normally manage credentials manually. From a user perspective, an increasing number of identifiers and credentials rapidly becomes totally unmanageable.

Some of the identity models described above, especially the federated model, have been motivated by the need to simplify the user experience. The idea is that if the user only needs to manage one set of identifier and credential, memorisation or other primitive methods for storing credentials are still acceptable. However, it is inconceivable that only one single federation domain will exist, and it is evident that there will never be a single identity domain for all service providers. Also, services with different levels of sensitivity and risk will require different types of credentials. As a rather optimistic scenario, it could be suggested that the number of identifier/credential sets a user needs to manage in case of widespread adoption of federated identity domains, would be 1 order of magnitude less than the number of service providers he accesses. Unfortunately, the user experience will still be poor when the number of online service providers is growing exponentially.

In our view, a totally new approach is needed. It seems natural to introduce automation and system support of the identity management at the user side. Expecting users to manage an unavoidably growing number of passwords and credentials by memorisation or other primitive methods is totally unrealistic.

A solution, which seems quite obvious, is simply to let users store identifiers and credentials from different service providers in a single tamper resistant hardware device which could be a smart card or some other portable personal device. This approach opens up a multitude of possibilities of improving the user experience and of strengthening the mutual authentication between users and service providers. Because its main purpose would be authentication, the device can be called a personal authentication device (PAD). This is illustrated in Fig.7 below.

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* [7]. Because the PAD is a personal device for identity management support, this architecture can be called user-centric identity management. It can be combined with any traditional identity management model described above, where Fig.7 represents an example illustrating how it can be combined with the isolated identifier domains of Sec.3.1

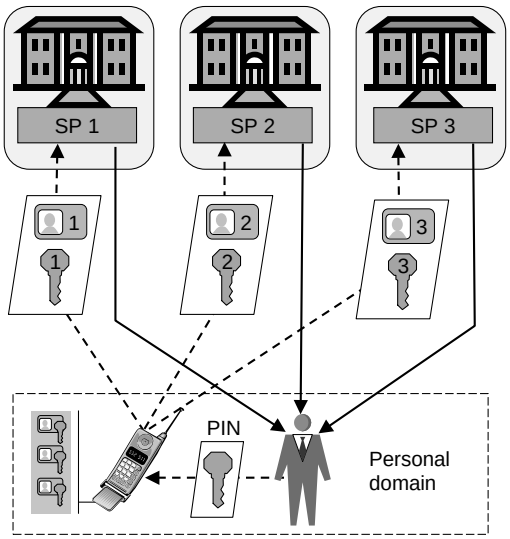

Figure 7: User-centric identity model.

The user must authenticate himself to the PAD, e.g. with a PIN, before the PAD can be used for authentication purposes. Many different authentication and access models can be imagined with a PAD. In case the PAD has a keyboard and display, a simple solution could for example be to retrieve from PAD memory a static password, or let the PAD generate

a dynamic password, that the user then can type into the login screen of the service provider. A more advanced solution could be to connect the PAD to the client platform via a communication channel such as bluetooth or wireless LAN, or to let the PAD communicate directly with the server through a secondary channel. This would allow the PAD to be fully integrated into the authentication process. This is described in more detail in Sec.6.

The functionality of a PAD could be integrated into other devices such as a mobile phone or personal digital assistant (PDA) which many people carry already. Using a mobile phone would also allow advanced solutions such as registration and challenge- response authentication through a mobile secondary channel. With a PAD connected to the client platform, virtual SSO solutions are possible. This could be implemented by letting the PAD automatically authenticate itself on behalf of the user as long as the PAD is connected to the client platform. The advantages of the user-centric user identity management architecture are that 1) the user only needs to remember one credential (e.g. the PAD PIN), 2) that virtual SSO is possible, and 3) that the traditional legacy identity management models described in Sec.3 can remain unchanged.

Signs of this type of solution are already emerging. For example, the Mozilla browser provides virtual SSO capabilities for users so they do not have to remember their usernames or passwords for web sites. A master password protects the PKCS11 security device, which can be either a software or hardware device that stores sensitive information associated with their identity, such as usernames and passwords, keys and certificates. Recent releases of Mozilla have a software-based security device, and can also use external security devices, such as smart cards, if the user's computer is configured to use them. The master password for the browser's built-in software security device protects the user's master key, which is used to encrypt sensitive information such as email passwords, web site passwords, and other sensitive data [9].

The PAD should be under the control of the user, and not under the control of the identifier providers, the credential issuers or the service providers. The latter would result in a proliferation of PADs which would defeat the purpose of having a single device for simplification identity management for the users. In order to gain full advantage of the PAD, it should be a general security device capable of handling many types of identities and credentials. Some level of standardisation, such as that described in the *Personal Transaction Protocol* [7], might be needed for that to be practical.

5 Management of Service Provider Identities

There are some fundamental differences between the management of user identities and service provider identities. Service providers usually have data registers of all their clients' identifiers and authorisation credentials linked to the authentication systems. The users, on the other hand, usually do not have a data register of the service providers they have a relationship with. It can therefore be difficult to determine the appropriate digital SP identifier that should be used to authenticate a given service provider.

Service providers that operate in a global environment such as the Internet need global identifiers. Unfortunately, there exists no reliable and practical global name space for people and organisations, so that it is questionable how meaningful service provider authentication really is in the current web security paradigm. Telephone numbers, email addresses, IP addresses, Internet domain names and OID actually represent global identifiers but because they often change, they can not be considered as stable and reliable identifiers for persons or organisations. There are examples of service provider identity domains with stable and reliable unique identifiers, but none of these identity domains are both global and comprehensive at the same time. National company registers used for tax purposes offer a comprehensive list of unique identifiers on a national level. The Australian Business Number Digital Signature Certificate [2] is an example of how this type of identity registers can be leveraged to allow strong authentication of or-

ganisations. The Dunn & Bradstreet company number register [1] offers a global list of unique identifiers, but unfortunately it is not comprehensive, and also lacks the character of being authoritative.

5.1 Common SP Identity Model

Despite the fact that no reliable global name space exists for service providers in general, cryptographically strong authentication solutions have been implemented, e.g. in the form of the Web PKI combined with the SSL security protocol.

Several identifiers, such as company name, street address, domain name etc., are encoded within Web PKI certificates used in SSL. The identifiers are sent as part of the server certificate in the initial phase of the SSL protocol. The user is unable to verify the credential himself, and relies on the computer to do it for him. On successful SSL verification of the credentials provided by the server, the client web browser displays a padlock in the corner of the browser window. This SP identity management model which is used by SSL is illustrated in Fig.8. SP identifier and credentials providers are commonly known as CAs in the SSL security model.

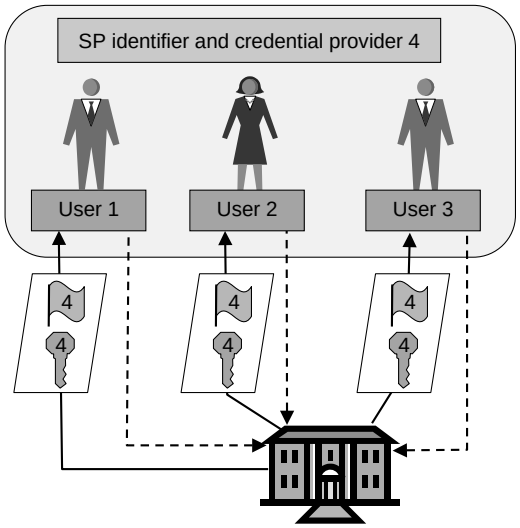

Figure 8: Common SP identity model.

A problem with the SSL security model is that the SP identity authenticated by the client browser not necessarily is the SP identity intended by the user.

The common scam called *password phishing* illustrates the difficulty of users to fully understand service provider authentication. There are also other ways to exploit the limitations of human cognitive power. One such example is *typo squatting* which consists of using domain names that are very similar to other domain names, for example differing only by a single letter so that a false domain name may pass undetected. How easy is it for example to distinguish between the following URLs: `http://www.bellabs.com`, `http://www.belllabs.com`, and `http://www.bell-labs.com`? Although strong cryptographic mechanisms are being used, it can be difficult for clients to know which identity has been authenticated by the browser.

This means that SSL performs SP authentication in a purely technical sense, but not in a semantic sense. SSL does however provide cryptographically strong confidentiality, and because SSL is widely used, it has resulted in the total elimination of the earlier practice of password sniffing that relied on passwords being sent in cleartext across the Internet.

When running SSL in so-called Anonymous Diffie-Hellman mode, it can provide cryptographically string confidentiality without PKI certificates. Because SSL authentication with certificates has limited value due to poor usability, and because confidentiality can be achieved without certificates, the value of using SSL certificates in SSL has very questionable value. By taking a more a user centric approach, we will in Sec.5.3 show how PKI certificates can be leveraged to provide more meaningful authentication of SPs. First we will describe the *Isolated SP Identity Model* which is practically unrealistic, but which can be simulated with a user centric approach.

[1] http://www.dnb.com/us/

5.2 Isolated SP identity Model

The isolated SP identity model emerges by turning the isolated user identity model of Sec.3.1 upside-down, as illustrated in Fig.9.

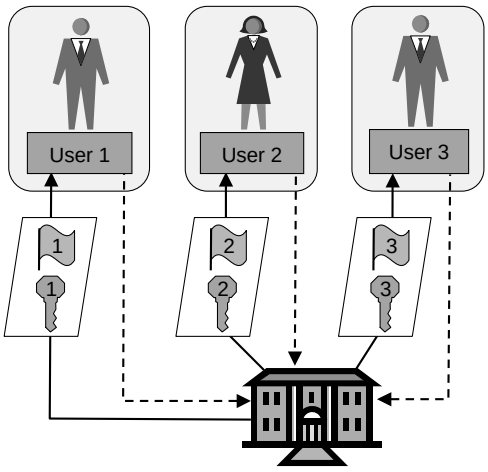

Figure 9: Isolated SP identity model model.

In this model, each user defines a personal name space for service providers, and assigns private identifiers to the service providers he or she wants to transact with. As a result, each SP must use different identifiers and credentials for itself when authenticating itself to different users. The indices of identifiers and credentials in Fig.9 relate to the user who assigned them. The advantage of this model would be that the personal SP identifiers are meaningful because they are assigned by the users themselves. However, it is quite obvious that this model is rather awkward, and that it would never work in practice. In the next section we will show how we can achieve a virtual isolated SP identity model by taking a user centric approach.

5.3 Personal SP Identity Model

Assuming that each user owns a PAD as described in Sec.4, 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.

The index "4" of the identifier and credential contained in the messages of Fig.10 indicates that they have been assigned by the centralised identifier and credentials provider with the same index, which in practice can be a CA of a PKI. We can thus assume that the messages contain PKI 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.

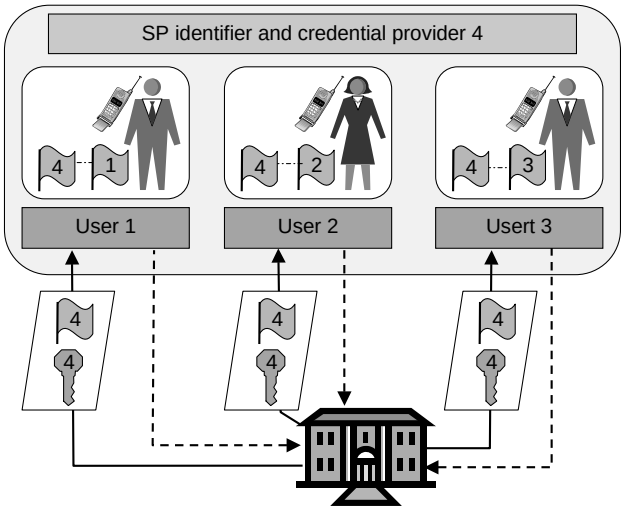

Figure 10: Personal SP identity model model.

The Mozilla TrustBar [4] is a current 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.4. 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, and the SSL padlock to 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.

DSTC's[2] prototype user-centric identity management solution, *Piccola*[3], aims to provide a management framework that allows users securely manage both the smart card its applications, in a vendor-independent way. It primarily targets Global Platform[4] (GP) compliant cards.

Global Platform is an initiative by the smart card industry to develop a universal hardware-neutral, vendor-neutral, application independent card management specification. It defines common security and card management frameworks, thereby providing a universal card platform for application developers and issuers. It attempts to address the security and management concerns related to each entity involved during the smart card life cycle.

While the Global Platform specification is technically sound and widely supported by major card vendors, there are various issues associated with it that hinder its uptake. Firstly, there is no publicly available reference implementation of the GP specification. Secondly, many vendor-supplied smart card development kits are often restricted to their own 'GP-compliant' cards, and therefore places limits on code reuse and interoperability. *Piccola* was created to address these shortcomings and more importantly can be used as a vehicle to develop smart card deployment solutions that are alternatives to the vendor-supplied proprietary isolated identity solutions. However, *Piccola* is not intended at competing with or replacing card vendors' software development kits. Instead, it allows for new business models and innovative methods of shared smart card application management, which can have enormous flow-on benefits for users.

6 Discussion

In Sec.4 and Sec.5.3 above, we described user centric approaches to user identity management and to SP identity management respectively. It is natural to combine these two aspects of identity management in order to provide a seamless user-centric system for two-way authentication. For example, the most typical case will be to have isolated user identity domains combined with a PKI based common SP identity domain. In this case, the PAD can link the SP certificate to the user identifier, as well as map it to the personal logo or image that the user has chosen as personal identifier for that service provider. This

[2] http://dstc.edu.au/

[3] http://titanium.dstc.edu.au/security/Piccola/

[4] http://www.globalplatform.org/

then represents the user's view of the world, i.e. his personal set of service providers with associated user identifiers for accessing them.

By implementing personal identity management in a separate device such as the PAD, many different authentication architectures become possible, which can be grouped into the main categories *single channel* and *dual channel* authentication, as illustrated in Fig.11 below.

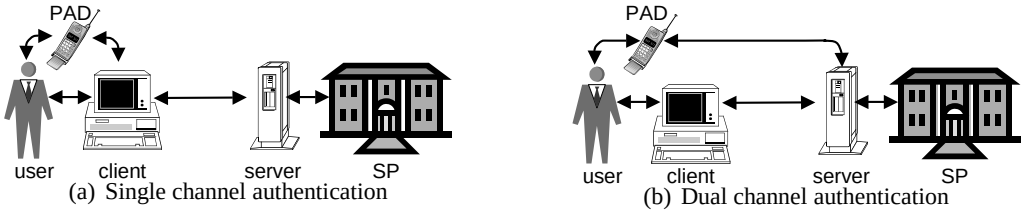

Figure 11: Possible authentication architectures when using a PAD.

Dual channel authentication protocols using mobile devices have been described in [3, 5], however they only describe dual channel user authentication, i.e. one-way authentication.

In order to provide a seamless user experience when doing two-way authentication, new two-way dual channel authentication protocols need to developed.

7 Conclusion

The characteristics of the user-centric identity management approach described in this paper can be summarised as follows.

- System supported identity management on the user side, resulting in improved usability.
- Protocol flexibility, by having a PAD that supports multiple authentication protocols and technologies.
- Mobility, by allowing the user to use any hardware platform when accessing online services, as long as he carries the PAD with him.
- Backwards compatibility, by not requiring replacement of legacy identity management systems.

In conclusion, we believe a user-centric approach to identity management is a very promising way to improving the user experience, and thereby the security of online service provision as a whole. This has the potential to stimulate increased uptake of online services.

References

[1] UK e Envoy. *Registration and Authentication.* UK Office of the e-Envoy, under the Cabinet Office, http://e-government.cabinetoffice.gov.uk/assetRoot/04/00/09/60/04000960.pdf, September 2002.

[2] National Office for the Information Economy. *Australian Business Number Digital Signature Certificate (ABM-DSC), Broad Specification.* NOIE (now AGIMO http://www.agimo.gov.au/), 2003.

[3] E. Gieseke and J. McLaughlin. Secure Web Authentication with Mobile Phones Using Keyed Hash Authentication. Technical report, Harward University, 11 January 2005.

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

[5] S. Lannerstrom. Mobile Authentication. Technical Report MPM 02:0041, SmartTrust/Sonera, 9 August 2002.

[6] 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.

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

[8] OASIS. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.* Organization for the Advancement of Structured Information Standards, OASIS Standard, 15 March 2005.

[9] Mozilla Project. *Privacy and Security Preferences - Web Passwords.*
URL: http://www.mozilla.org/projects/security/pki/psm/help_20/passwords_help.html, 2004.

[10] Shibboleth Project. *Shibboleth Architecture Protocols and Profiles.* Working Draft 05, 23 November 2004. Internet2/MACE, 2004.