Portable User-Centric Identity Management

Gail-Joon Ahn, Moo Nam Ko and Mohamed Shehab

Abstract User-centric identity management has recently received significant attention for handling private and critical identity attributes. The notable idea of user-centric identity management allows users to control their own digital identities. Current user-centric identity management approaches are mainly focused on interoperable architectures between existing identity management systems. Normally, users can access the Internet from various places such as home, office, school or public Internet café. We observe that the importance of portability of the a user's digital identity should be addressed in the user-centric identity management practices. In other words, users should be able to export their digital identities and transfer them to various computers in a secure manner. In this paper, we focus on the portability issue of the Identity Metasystem and describe three possible types of portability-enhanced Identity Metasystem model including our implementation experience.

1 Introduction

The Internet has dramatically changed the way people communicate and do business. Businesses heavily depend on the Internet to draw in commerce and make information available on demand. Managing bank accounts, paying bills and purchasing goods via Internet are commonly exercised. The diverse Internet services

Gail-Joon Ahn
Arizona State University, Ira A. Fulton School of Engineering, Department of Computer Science and Engineering, 699 S. Mill Avenue, Tempe, AZ 85281, e-mail: gahn@asu.edu

Moo Nam Ko
The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, e-mail: mnko@uncc.edu

Mohamed Shehab
The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, e-mail: mshehab@uncc.edu

Please use the following format when citing this chapter:

Ahn, G.-J., Ko, M.N. and Shehab, M., 2008, in IFIP International Federation for Information Processing, Volume 278; *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 573–587.

and the tremendous amounts of personal data collected over the Internet have raised various problems such as identity theft, fraud, and privacy breaches [22]. Numerous identity management systems have been introduced to solve the identity management problems of business domains [1]. Different identity management systems have their strengths and weaknesses and have been deployed in different contexts. Most identity management systems were designed mainly from the business's perspective. Users were not considered carefully in the design stage which led to serious identity related problems. In addition, most identity management systems have focused on identity management issues in an isolated domain and federation issues between identity management systems in the circle of trust.

The digital identity industry recognizes that identity management systems are designed without the consideration of user experience and the non-interoperability between current identity management systems which restricts the growth of e-commerce activities. As a result, user-centric identity management has recently received significant attention for handling private and critical identity attributes. The main objective of user-centric identity management is to put the users in control of their identity information. Users are allowed to select their credentials that are used to respond to an authentication or attribute requester. Through the user-centric identity management, the users have more rights and responsibilities for their identity information than before. In this paper, we articulate the portability issues of the user-centric identity management system, attempting to enhance an existing Identity Metasystem. The paper is organized as follows. Section 2 overviews the digital identity management and discusses the related technologies. Section 3 describes our portability enhanced Identity Metasystem approaches. Section 4 describes implementation details followed by the related works in Section 5. Section 6 includes the concluding remarks.

2 Digital Identity Management

In this section, we first start with the discussion of digital identity and digital identity management. We then discuss the user-centric identity management approach, portability issues and the related technologies.

2.1 Digital Identity

There are various definitions of digital identity. Depending on organizations, systems and contexts, the diverse definitions of digital identity have been created and used. From our perspective, we define a user's digital identity as the global set of attributes that make up an online representation of who and what an entity is. It can

[1] Such identity management systems include IBM Tivoli [11], Liberty Alliance [18], LID [19], OpenID [24], Sxip [31], Microsoft CardSpace [40] and Live ID [41]

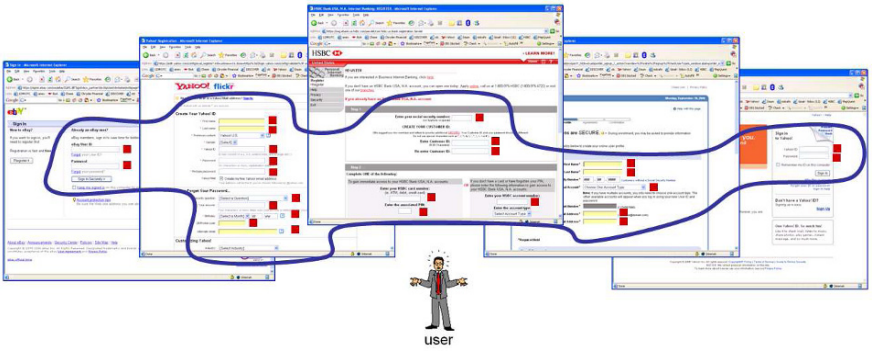


Fig. 1 Digital Identity: Global Set of Attributes of a User

include access credentials, personal attributes and personal references. Over the Internet, a user has numerous access credentials that are issued from different sites and different or duplicated personal attributes and references on each site. We believe all of these attributes should be considered as the user's digital identity as shown in Figure 1. In each site, a user can be represented by subsets of these attributes. Depending on the situation and the context, different subsets of attributes are used to represent the same user in the Internet. For example, in an auction site, a subset of a user's attributes such as username, password, shopping history, and reputation record represent the user's identity in this site, while a subset of the user's attributes such as a student ID number, class record, and GPA may represent the user's identity in an university site.

2.2 Digital Identity Management

Digital identity management consists of several tasks such as maintaining user attributes and using subsets of attributes to enable secure online interactions between users or between users and systems. Digital identity management enables the addition, utilization, and deletion of identity attributes. In [2], the identity management systems are categorized into three models: isolated, centralized, and distributed identity management. In the isolated identity management model, each site has its own identity management domain and its own way of maintaining the identities of users including employees, customers, and partners. The centralized identity management model has a single identity provider that brokers trust to other participating members or service providers in a circle of trust. The distributed identity management model provides a frictionless identity management solution by forming a federation and making authentication a distributed task. Every member agrees to trust user identities vouched for by other members of the federation. These identity management models were mostly focused on the domain centric approach. Our analysis and observation indicate that most identity management systems neglect

user-friendliness and usability issues. Therefore, it leads users to be the weakest link in digital identity management systems.

2.3 User-Centric Identity Management and Portability

Under domain centric identity management systems, a user's information is collected and managed by service providers so it is difficult for the user to manage their identity information located at service providers and to monitor the usage of the user's private information. Putting the owner of the identity information into the transaction gives the user-centric identity management approach the ability to solve identity related problems. To achieve the goal, several requirements from the user's perspective need to be accommodated in the design of user-centric identity management systems. As the users have more rights on their own identity information, they can decide what information they want to share, how much information to be disclosed with other trusted service providers, and under what circumstances. Thereby better protection of the user's private information is enabled by user.

Domain centric identity management systems focus on the user authentication to protect their properties from malicious users. However, the authentication of service providers is equally important for a user to figure out the trustworthiness of the service providers. Current browsers provide the padlock icon to give notice to the users for the SSL communication between the users and service providers but it is not enough for the users to figure out the trustworthiness of the service providers [44]. By providing the identity information of service providers clearly to the users in web-based interactions enables the users to distinguish trusted service providers from malicious service providers. The users can then decide to disclose their information to only trusted service providers. Hence, the users can protect their information from phishing attacks and possible frauds.

In the current Internet environments, a user has to create a separate account for each web site the user wishes to access. The user also has to maintain multiple separate accounts, which would be a tedious job. In addition, the users often choose insecure passwords, rarely change their passwords, and use the same password across different accounts [1]. These trends make the password-based authentication systems insecure. New strong authentication methods are required to overcome the security problems of the password-based authentication method. The new methods should be easy for the users to manage their digital identities. Existing identity management systems provide different user experiences and interfaces that could lead the users to improperly interact with different entities in Internet environments. Under the user-centric identity management systems, the users manage their identity information directly through a proper interface which provides a consistent experience to control their identity information legitimately.

People carry identity cards such as a driver license card, a student ID card, and an employee ID card in their wallet and they use each identity card in its appropriate context. Similar to the identity cards in the real world, the digital identity should

be carried by the users and it should be used without the limitation of locations and devices. Actually, people access the Internet from different sites such as home, office, school, public Internet café, and so on. Therefore, the digital identity should be both interoperable and portable.

2.4 Related Technologies

The Identity Metasystem is an interoperable architecture for digital identity management [6]. It is defined based on the "Laws of Identity" which are intended to codify a set of fundamental principles to which any universally adopted, sustainable identity architecture must conform [5]. The Identity Metasystem provides interoperability between existing and future identity systems using Web Services (WS-*) protocols which is a set of specifications built on the web service platform. Specifically, WS-Trust [38], an encapsulation protocol, is used for the claim transformation. WS-MetadataExchange [35] and WS-SecurityPolicy [37] are used to conduct the format and claim negotiations between participants. Finally, WS-Security [36] is used to secure transmitted messages. The Identity Metasystem can transform the claims of one type into the claims of another type and WS-* protocols negotiate the acceptable claim type between two parties to provide interoperability between them. The Identity Metasystem also provides a consistent and straightforward user interface to all the users. There are three roles within the identity metasystem: *Identity Providers* who issue digital identities, *Relying Parties* who require identities, and *Subjects* who are individuals and other entities about whom claims are made. To build an identity metasystem, the system is required to follow five key components [22]:

1. A way to represent identities using claims.
2. A means for identity providers, relying parties, and subjects to negotiate.
3. An encapsulating protocols to obtain claims and requirements.
4. A means to bridge technology and organizational boundaries using claims transformation.
5. A consistent user experience across multiple contexts, technologies, and operators.

CardSpace [40], is an implementation of the Identity Metasystem, provides the consistent user experience required by the Identity Metasystem. When a user needs to authenticate to a relying party, CardSpace interprets the security policy of the relying party and displays an Identity Selector containing a set of information cards which satisfy the requested claims in the relying party's security policy. Once the user selects a card, CardSpace contacts the relevant identity provider and requests a security token. The identity provider generates a signed and encrypted security token which includes the required information and returns it to the Identity Selector. The user then decides whether to release this information to the relying party. If the user approves then the token is sent to the relying party where the token is processed and the user is authenticated.

Java Card is a Smart Card running a small Java based operating system. It is useful in the areas of personal security and can be used to add authentication and secure access to information systems that require a high level of security. The user can carry around valuable and sensitive personal information such as medical history, credit card numbers and private key in the Java card. The Java Card technology enables Smart Cards and other devices with very limited memory to run small applications (applets) and provides Smart Card manufactures with a secure and interoperable execution platform that can store and update multiple applications on a single device [14]. Java-powered iButton is based on Java Card technology and provides the processing features which include a high-speed 1024 bit RSA encryption, Non-Volatile RAM(NVRAM) capacity, and unalterable realtime clock [8]. It utilizes NVRAM for program and data storage. Unlike electrically erasable programmable read-only memory, the NVRAM iButton memory can be erased and rewritten as often as necessary without wearing out. Therefore multiple applets can co-exist in NVRAM and control the sensitive data in a secure way. It can be attached to accessories such as a key fob, watch, and finger ring so the users can easiliy carry the iButton. We adopt this technology to demonstrate the feasibility of our approach.

3 PORTABILITY IN IDENTITY METASYSTEM

In this section we discuss the principles behind the Identity Metasystem and seek methods to extend the Identity Metasystem for addressing portability aspects. We focus on the information card and security token service modules in the Identity Metasystem.

3.1 Information Card

The users of Identity Metasystem can manage their digital identities using visual information cards in the Identity Selector. The information card draws a line between the self-issued card and the managed card. Both types of information cards do not contain personally identifiable information (PII). The information card generally contains the card name, card image, a list of claims, and issuer information. However, there are differences between the two types of information cards. In case of the self-issued card, after the user provides the general user's information such as last name, first name, and e-mail address, the Identity Selector grants the user a self-issued card. The self-issued card is stored in the local machine. Although the self-issued card includes general PII, it is not supposed to include the sensitive user information such as social security number, bank account and credit card number. On the other hand, the managed cards are obtained from identity providers such as employers, financial institutions, or the government. Like the self-issued information card, the managed card can be stored in local machines but the PII associated

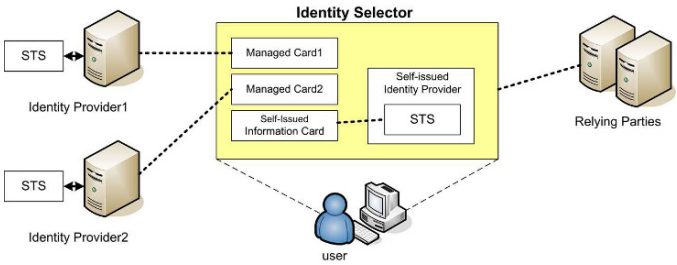


Fig. 2 STSs and Information Cards in Identity Metasystem

with the card is not stored in the local machine. The PII is stored and managed by each identity provider. The managed card enables the identity providers to issue their own set of claims. For example, credit card companies can design a set of claims such as card name, card number and expiration date in their managed card and the DMV can design a set of claims such as driver license number, license class and expiration date in their managed card.

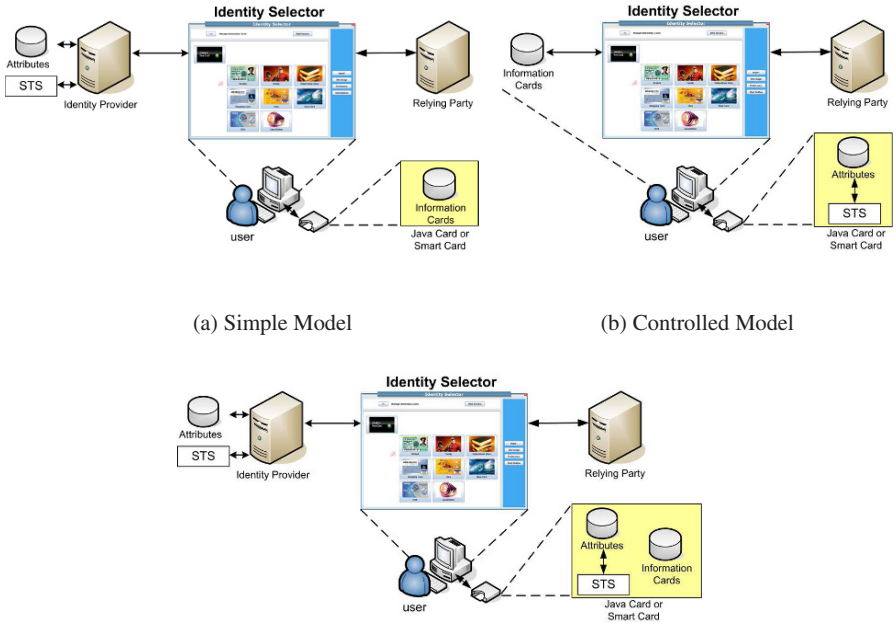
3.2 Security Token Service

When the digital identities are transmitted on the network, every digital identity is presented by some sort of security tokens such as X.509 certificate [42], Kerberos ticket [16], and SAML assertion [28]. The Identity Metasystem generates a security token by contacting the Security Token Service (STS) in the identity provider. When the Identity Selector sends a "RequestSecurityToken" message to the identity provider, the STS in identity provider responds back with a "RequestSecurityTokenResponse" message that contains a new security token. The current implementation of Identity Metasystem has two STSs as illustrated in Figure 2. The STS located at the third party identity provider generates security tokens for the managed cards, whereas the STS in Identity Selector at the user's local machine generates the security tokens for the self-issued cards.

3.3 Portability of Information Cards and STS

The CardSpace stores the information cards in a local machine and provides basic import and export functions for information cards. Using these functions, the users can export their information cards to portable storage devices such as portable USB flash drive, mobile phone, and PDAs and import the information cards into other machines. When the information cards are exported, the information cards are encrypted using a key derived from a user-selected pass-phrase [7]. Hence, if a user

loses a portable storage device with the exported information cards, other people cannot decrypt the exported information cards unless they know the pass-phrase of the information cards. However, these export and import functions are not sufficient to support the various practical scenarios. For example, a user carries the exported information cards in a USB flash driver and imports the information cards in a kiosk machine from the USB flash driver. After using the information cards in the kiosk machine, if the user forgets to delete the imported information cards, then the next user of the kiosk machine can access the previous users' information cards without any restrictions. The bottom line is to enable the users to carry the information cards in a secure manner, considering the portability of STS as well. To achieve such an intrinsic goal, we categorize the portability enhanced Identity Metasystem into three models based on the location of the information cards and STS as follows:



(a) Simple Model

(b) Controlled Model

(c) Integrated Model

Fig. 3 Portability-enhanced User-centric Identity Management

- ***Simple Model:*** This model is similar to the general architecture of the Identity Metasystem. Figure 3(a) shows the simple portability enhanced Identity Metasystem model. The STS is located in the identity provider and the users carry their information cards using portable secure devices such as Java Card or Smart Card. By storing the information cards in portable secure devices, only a user who knows the PIN number of the secure device can access the information cards

and is able to export their information cards to multiple machines. When the user removes the secure device from a machine, the imported information cards should be removed from the machine automatically. This model can be applied between different machines to synchronize the information cards.

- ***Controlled Model:*** This model shifts the role of identity provider to the portable secure device. The user's attributes and STS are located in a portable secure device and the information cards are located in a local machine. The user carries the STS and attributes in portable secure devices so the Identity Selector does not have to contact the identity provider to get a secure token. The Identity Selector directly contacts the STS in the portable secure device and gets the security token. The Figure 3(b) shows the controlled portability enhanced Identity Metasystem model. This model can be applied to the one-time credit number system [4, 39], where A credit card company issues a portable secure device with STS to the customers. The customers can treat the portable secure device with STS as a portable identity provider. When a customer does an online purchase, the Identity Selector gets a secure token from the STS in the portable secure device directly. The issued secure token includes the one-time credit card number so the user can protect the real credit card number. The drawback of this model is that information cards are still in a local machine and a high expense is expected to distribute the portable STS devices
- ***Integrated Model:*** This model is a combination of the Simple and Controlled models. The users carry the information cards, STS and attributes in a portable secure device, this enables them to directly manage their identity. When a user plugs a portable secure device into a machine and provides the PIN number, the identity selector imports and shows the information cards in a portable secure device to the user. After the user selects a managed information card which requests a token from the STS in the portable secure device, the Identity Selector directly gets the request token from the STS in the portable secure device. This model combines the advantages of previous models so the user can carry their information cards, portable STS and attributes in a secure device according to the user's purpose. This model gives the user more flexibility and extensibility to manage his/her digital identities. Figure 3(c) shows the integrated portability enhanced Identity Metasystem model.

4 Implementation Details

Based on our analysis of the Identity Metasystem and articulation of potability-enhanced models, we developed a prototype of the Identity Selector, which is a Java-based implementation of Identity Metasystem. Furthermore, we enhanced the Identity Selector to support our portability enhanced Identity Metasystem models. In this section we give an overview of our implementation experience and outcomes.

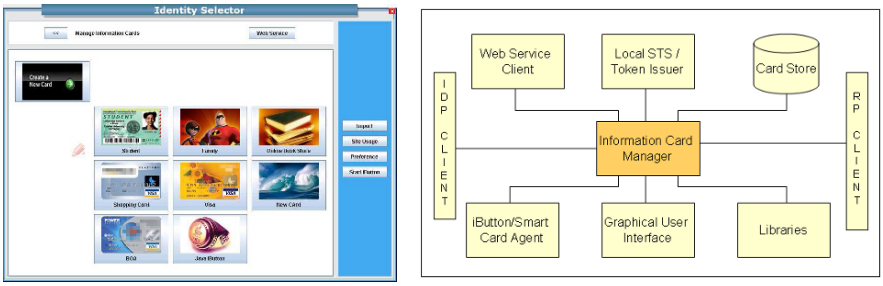
4.1 Identity Selector

Identity Selector is an important component in Identity Metasystem. Using the visual information card, the users can select their identity cards with the same experience as the one in their real life. Figure 4 illustrates our Java-based prototype of CardSpace-compatible Identity Selector. Each information card contains a subset of the available user attributes that are used to represent the user's identities in different contexts. Each card mainly includes meta information required to acquire the real attributes from the identity provider. The meta information includes the necessary user attribute fields, identity provider contact information, and token related information.

Our Identity Selector consists of seven components: Information Card Manager, Graphical User Interface, Card Store, iButton/Smartcard Agent, Web Service Client, Local STS/Token Issuer, and libraries as shown in Figure 4 (b). The Information Card Manager handles all events generated by users and systems, and performs the appropriate action. It also provides the card creation, editing, and deleting functions for the self-issued information card. The Graphical User Interface component manages the user interface of Identity Selector. It consists of a set of screens such as the creation of new card, the examination of cards and the selection of a card. The Card Store contains information cards, which are usually stored in XML format. The Web-Service Client supports the communication between identity provider and Identity Selector. The iButton/Smartcard agent manages the communication between the Identity Selector and the Java-powered iButton. It sends the PIN number and token request message to iButton and receives the issued token from iButton. The iButton/Smartcard agent and the Java-powered iButton exchange messages using the APDU (Application Protocol Data Unit). The Java-powered iButton includes the Java Applet which provides STS module, user attribute storage, and information card storage. The Java Applet is designed based on our integrated model. The Local STS/Token Issuer generates CardSpace compatible security token for self-issued information card and also transforms the token issued from iButton to the CardSapce compatible security token. Using openSAML 1.1 [25], Bouncy Castle API [32] and our libraries, the local STS/Token Issuer encrypts and signs the XML token. The libraries include the required standard and customized modules that are necessary for supporting the functionalities of Identity Selector.

4.2 Portable Security Token Service

To generate a CardSpace-compatible security token in portable secure devices, the Portable Security Token Service (PSTS) needs to support strong cryptographic algorithms. Moreover, portable secure devices should be able to generate SAML asser-



(a) Interface (b) Components

Fig. 4 Java version of Identity Selector

tions. We identify three approaches to address how CardSpace-compatible security tokens can be generated by Java Card technology [2].

- *Basic Mode*: The PSTS in Java Card generates its own token and the local STS in Identity Selector transforms the issued token into a CardSpace compatible security token. The local STS signs and encrypts the token for the relying party. This PSTS approach is only available for self-issued cards.
- *Non-auditing Mode*: The PSTS in Java Card generates a SAML assertion and then the local STS in Identity Selector encrypts it for the relying party. This is a "non-auditing" mode of Identity Metasystem [4], as the identity provider has no knowledge of the relying party to protect the user's privacy for Internet activities. In other words, when Identity Selector receives a singed token from Identity provider, PSTS can generates the SAML assertion by using a predefined XML SAML assertion document and dynamically generated assertion data such as digested value, signature values, and RSA public key value. Identity Selector then encrypts the SAML assertion for the relying party. This approach can be applied to both self-issued information cards and managed information cards.
- *Auditing Mode*: The PSTS in Java Card directly generates CardSpace compatible security token for the relying party under the assumption that Java Card supports the WS-Trust standard with strong cryptographic algorithms. When the PSTS generates the security token, the PSTS knows the identity of relying party and generates the security token for relying party directly. This is in "auditing" mode of Identity Metasystem [4]. When PSTS receives "RequestSecurityToken" message from Identity Selector, the PSTS generates a security token for the relying party and sends it to Identity Selector using "RequestSecurityTokenResponse" message. This approach is similar to current .NET Smart Card approach and it is

[2] .Net Smart Cards such as Gemalto Cryptoflex NET [9] and MXI security Stealth MXP [30] can also provide cryptographic functions necessary to implement the PSTS.

can be easily implemented when Java Card supports the WS-Trust standard with strong cryptographic algorithms.

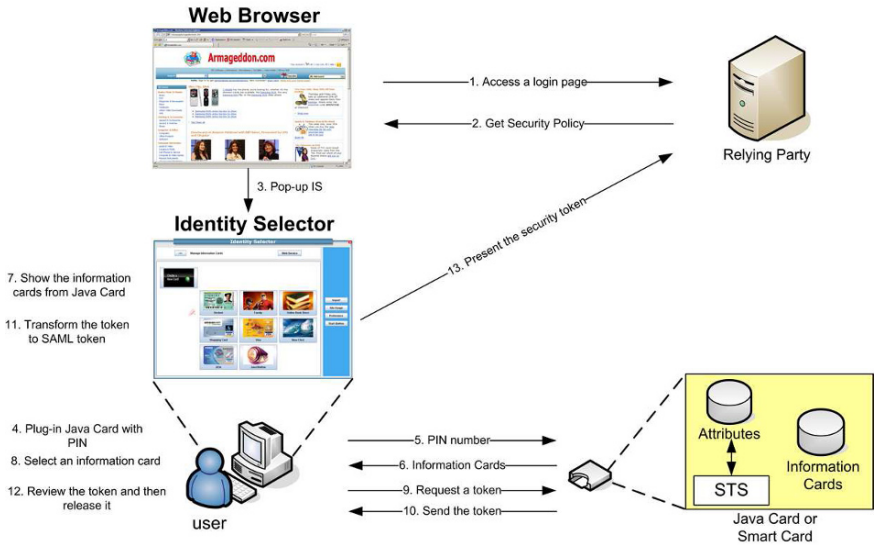


Fig. 5 System Flows and Corresponding Messages

Our prototype of the PSTS applet and iButton/ SmartCard agent is based on the *Basic Mode*. Using a predefined protocol, iButton/ SmartCard agent requests a token for self-issued card to PSTS applet. The PSTS applet is a PIN protected applet and provides card storage, user attribute storage, and token generation service.

Figure 5 depicts the system flow diagram and corresponding messages in our portability-enhanced user-centric identity management model. The process begins when a user accesses a login page at a relying party's web site. The site sends a login form to the browser. The login form contains a specific OBJECT tag which includes the site's security policy and invokes the Identity Selector, which displays the information cards that satisfy the relying party's security policy. On the other hand, when the user accesses a kiosk machine, the Identity Selector does not contain any cards because the kiosk machine should not store the user's information cards. In that case, the user needs to select the iButton mode and insert a Java-Powered iButton into the kiosk machine. The iButton agent in Identity Selector immediately recognizes the iButton and asks for the PIN to reads the information cards from iButton. Next, the user selects an information card and the Identity Selector sends the token requests to iButton. The Identity Selector transforms the token issued by iButton into a CardSpace compatible security token using the local STS module and displays the attribute information. If the user consents to release the security token, the Identity Selector presents the security token to the relying party. Finally, the relying party verifies the security token as part of the authentication process. With

this scenario, we believe our prototype enable users to carry their digital identities using portable secure devices.

5 Related Works and Discussion

There are several open source projects for user-centric identity management systems or related technologies. To address the interoperability issue among those identity management systems, the Open-Source Identity Systems (OSIS) working group was formed [33]. The OSIS fosters several identity-related open-source projects such as Bandit [3], Heraldry [12], Higgins [10], OpenSSO [26], OpenXRI [27], Shibboleth [29], and xmldap [43] and harmonizes the construction of an interoperable identity layer for the Internet.

In [23], the authors pointed out the portability problem of client side storage of user profile information. Once the user stores their information in a local machine, it assures that the user has as much control over their information as possible. However, the personal information stored on a local machine is not portable. The authors briefly suggested smart card or other portable devices to solve the portable problem in client side storage of user information. Another approach is to use IDRepository [17], IDRepository approach is to separate user profile information from the services, and store the identity in a central place where it can be maintained and accessed by appropriate entities. In [15], the authors allowed users to store identifiers and credentials from different service providers in a personal authentication device (PAD). The functionality of a PAD could be integrated into other approach.

In our work, the secure channel between smart card and smart card application and the trust of client machine might be issues in using portable secure device on various machines. Our approach assumes both secure channel and trustworthiness are intact. If the communication channel between smart card and smart card application is not secure, the communication can be monitored by malicious software on client machine. Markantonakis et al. [20] proposed a secure channel protocol between smart card and smart card application using the Diffie-Hellman protocol [34]. Using their approach we can further establish a secure channel between Identity Selector and Java Card as needed. In case of CardSpace, it runs on Secure Desktop in .NET Framework 3.0 [21] for preventing any distrusted activities in a client machine. To support this security feature, we would require trust computing technologies that can be either software or hardware-based solutions. These issues are currently being explored as ongoing research tasks.

6 Conclusion and future work

In this paper, we have articulated three types of portable Identity Metasystem models and explored the applicable environments of each model. To demonstrate our

models, we have developed our own prototype of a CardSpace-compatible Identity Selector using the Java language and extended the portability using Java Card technologies. We also proposed three possible approaches to generate CardSpace compatible security tokens using the Java Card. We believe our implementation demonstrated the feasibility of proposed portable user-centric identity management models that effectively enable the users to carry information cards and user attributes in a secure manner.

Our future work would include possible enhancements of our Identity Metasystem to support Web 2.0. Mashups and Social network service environments.In these environments users can share their information attributes with other users more frequently and easily through creative and innovative Web 2.0 based applications. Also, our work would include the development of metrics to characterize and measure user-centricity in the digital identity management that eventually leads us to have the common understanding of principles and practices. In addition, we strongly believe that private and critical identity attributes exchanged in our portable user-centric identity management models should be also protected based on the users' preferences. Such privacy-preservation techniques will be studied as part of our future works.

References

1. Adams, A. and Sasse, M. A. 1999. Users are not the enemy. Commun. ACM 42, 12 (Dec. 1999), 40-46. DOI= http://doi.acm.org/10.1145/322796.322806
2. Ahn, G. and Lam, J. 2005. Managing privacy preferences for federated identity management. In Proceedings of the 2005 Workshop on Digital Identity Management (Fairfax, VA, USA, November 11 - 11, 2005). DIM '05. ACM, New York, NY, 28-36. DOI= http://doi.acm.org/10.1145/1102486.1102492
3. Bandit-project.org Home. Available at http://www.bandit-project.org/
4. Cameron, K.: Kim Cameron's Identity Weblog. Available at http://www.identityblog.com/
5. Cameron, K.: The Laws of Identity. Microsoft Corporation, White Paper, May 2005
6. Cameron, K. and Jones, M.: Design Rationale behind the Identity Metasystem Architecture. Microsoft Corporation, White Paper, May 2005
7. Chappell, D.: Introducing InfoCard. Microsoft Corporation, Draft version for MIX, March 2006
8. Curry, S.:An introduction to the Java Ring, Java Wrold, April 1998
9. Gemalto Cryotoflex.NET. Available at http://www.cardsolutions.se/Cryptoflex.NET.pdf
10. Higgins Trust Framework Project Home. Available at http://www.eclipse.org/higgins/
11. Identity Management solutions from IBM Tivoli software. Available at http://www-306.ibm.com/software/tivoli/solutions/identity-mgmt/
12. Incubation Status for Heraldry. Available at http://incubator.apache.org/projects/heraldry.html
13. Java Card Technology. Available at http://java.sun.com/products/javacard/index.jsp
14. Java Card Technology Overview. Available at http://java.sun.com/products/javacard/overview.html
15. Jsang, A. and Pope, S.: User Centric Identity Management. Proceedings of AusCERT, Gold Coast, May 2005
16. Kerberos Token Profile 1.1. Available at http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf
17. Koch, M.:Global Identity Management to Boost Personalization, 9th reserch sysmp. on Emerging Electronic Markets, 137-148, 2002

18. Liberty Alliance Project. Available at http://www.projectliberty.org/
19. LID Wiki. Available at http://lid.netmesh.org
20. Markantonakis, K. and Mayes, K.: A Secure Channel Protocol for Multi-Application Smart Card Based on Public Key Cryptography, IFIP CMS, 2004
21. Microsoft .NET Framework 3.0 Community (NetFx3). Available at http://www.netfx3.com/
22. Microsofts Vision for an Identity Metasystem. Microsoft Corporation, White Paper, May 2005
23. Mulligan, D. and Schwartz, A. 2000. Your place or mine?: privacy concerns and solutions for server and client-side storage of personal information. In Proceedings of the Tenth Conference on Computers, Freedom and Privacy: Challenging the Assumptions (Toronto, Ontario, Canada, April 04 - 07, 2000). CFP '00. ACM, New York, NY, 81-84. DOI= http://doi.acm.org/10.1145/332186.332255
24. OpenID: an actually distributed identity. Available at http://openid.net/
25. OpenSAML - an Open Source Security Assertion Language toolkit. Available at http://www.opensaml.org/
26. OpenSSO Home, Available at https://opensso.dev.java.net/
27. OpenXRI.org Home. Available at http://openxri.org/
28. SAML Token Profile 1.1. Available at http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf
29. Shibboleth Project- Internet2 Middleware. Available at http://shibboleth.internet2.edu/
30. Stealth MXP. Available at http://www.mxisecurity.com/docs/mxi_stealth_mxp.pdf
31. Sxip identity. Available at http://www.sxip.com/
32. The Legion of the Bouncy Castle. Available at http://www.bouncycastle.org/
33. OSIS: Open Source Identity Systems. Available at http://osis.idcommons.net/
34. Ueli M. Maurer, Stefan Wolf: The Diffie-Hellman Protocol. Des. Codes Cryptography 19(2/3): 147-171 (2000)
35. Web Services Metadata Exchange(WS-MetadataExchange). Available at http://specs.xmlsoap.org/ws/2004/09/mex/W S-MetadataExchange.pdf
36. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). Available at http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
37. Web Services Security Policy Language(WS-SecurityPolicy). Available at http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf
38. Web Services Trust Language (WS-Trust). Available at http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf
39. What is ShopSafe. Available at http://www.bankofamerica.com/creditcards/index.cfm?template=faq
40. Windows CardSpace. Available at http://cardspace.netfx3.com/
41. Windows Live ID. Available at https://accountservices.passport.net/ppnetworkhome.srf?lc=1033
42. X.509 Token Profile 1.1. Available at http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf
43. xmldap.org - cardspace/infocard resources. Available at http://xmldap.org/
44. Ye, Z. and Smith, S. 2002. Trusted Paths for Browsers. In Proceedings of the 11th USENIX Security Symposium (August 05 - 09, 2002). D. Boneh, Ed. USENIX Security Symposium. USENIX Association, Berkeley, CA, 263-279