

User Control Enhanced Digital Identity Interchange Framework

Sang Rae Cho*, Seung-Hun Jin**

*ETRI (Electronics and Telecommunications Research Institute, Rep. of Korea)

**ETRI (Electronics and Telecommunications Research Institute, Rep. of Korea)

sangrae@etri.re.kr, jinsh@etri.re.kr

Abstract— This paper proposes a framework that covers how global interoperable digital identity interchange can be achieved and how user's privacy is enhanced by providing a user more control over the process of identity interchange.

Keywords— Digital Identity Management, User-Centric Identity Management, Authentication, Privacy Protection

I. INTRODUCTION

The federation identity has been introduced to connect distributed account information between an IdP (Identity Provider) and a SP (Service Provider). If the SP wants to assure authentication information from the IdP, there has to be a trust relationship between two parties. We call this trust domain Circle of Trust that consists of one or more IdPs and several SPs [1].

The concept of CoT (Circle of Trust) has a problem if the number of CoT grows exponentially. In this situation a user has to authenticate himself to a CoT every time he visits. In order to solve the problem, the identity federation between each CoT is necessary to connect all IdPs in every CoT. However the solution will be too complicated to deploy and manage [3][5].

It appears at first sight that the identity management with federation does not have any drawbacks. However the federation between entities needs prior business agreement. If the domain for identity management is limited within enterprise environment, then the federation technology works fine. But if the domain of IdM system extends to the Internet, it is difficult to reach business agreement between enterprises for all federations.

Discovery for identity service within a domain is not easy. Furthermore, discovery for inter-domain identity service is extremely difficult task to resolve and complex to manage.

Since identity information flows back and forth between enterprise entities without user's intervention, privacy control and consistent privacy protection can cause huge overhead for a system administration. The problem has arisen based on the fact that two entities try to share user's identity information, which belongs to the user. Since the two entities deal with the user's identity, they need to have

business and privacy policy agreement. If the one entity gives the user's identity to the user and the user can use it wherever he wishes, then each entity only needs to make a contract with the user for the use of his identity information [2].

In order to solve the problem, the paper defines a framework that covers how global interoperable digital identity interchange can be achieved and how user's privacy is enhanced by providing a user more control over the process of identity interchange.

II. CONCEPTUAL MODEL

A. Digital Identity Client

In this conceptual model, Digital Identity Interchange mechanism employs the concept of a digital identity client that can control the interchange of digital identity. Identity Interchange layer can also provide transparent identity interchange link layer to facilitate identity interchange between entities and to give a user full control to enforce his security and privacy policies.

The figure 1 illustrates the conceptual model for digital identity interchange. In the figure, Identity Web Server (IDWS) is the main server that provides user's identity information to an entity that requests. IDWS can be an identity provider if it mainly provides identity information; otherwise it can be an identity consumer. A web server can request user's identity to an identity web server to provide a web service.

A user in the conceptual model usually refers to a human being in user-centric IdM (Identity Management) context. However the concept of the user can easily extend to a device or an entity in network-centric environment.

A DIC (Digital Identity Client) is the system that manages user's identity and credential information. It enables the user to select his identity to share with various entities under user's control. DIC links every IdP and IdC (Identity Consumer) that the user wishes to share his identity. In both IdP and IdC, an Identity Web Server is deployed to establish and maintain a link with the DIC and provides identity information that is requested by DIC. The DIC is a client program that is installed in a user's computer to provide authentication and credential management, identity interchange and privacy protection service to the user. If DIC need to share identity information with IdP or IdC, then it establishes a link with an IDWS in a

domain. DIC may make a contract with IDWS to describe terms and conditions for identity interchange service to enforce privacy protection.

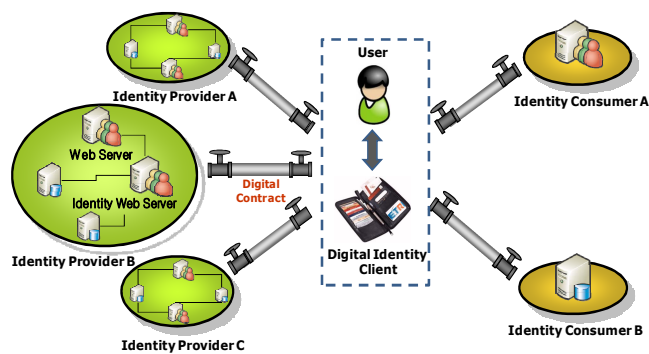


Figure 1. The Conceptual Model for Digital Identity Interchange

Every user’s identity information flows through DIC so that the user can control sharing of his identity information. Discovery of user’s identity information is not necessary since the client has all the link information that is required to push and pull identity information.

B. Digital Contract

Identity information that flows through between an identity provider and an identity consumer must go through a digital identity client. This gives a user an opportunity to control over his identity information. Digital Contract is the core component that can provide a fine-grained control for the user to control his identity information flows. The types of controls that digital contracts can define are any policy necessary to mediate an identity interchange relationship. The digital contract is not required on every identity interchange. It is only needed when it is necessary to control the flow or caching of a shared identity. This component is as flexible and extensible as real world contracts (e.g., non-disclosure agreements). Furthermore, since digital contracts can be themselves XML documents, they can govern their own revision, amendment, and deletion (like real-world contracts).

The figure 2 illustrates the structure of a digital contract. The structure is as flexible as possible to be applied to various circumstances.

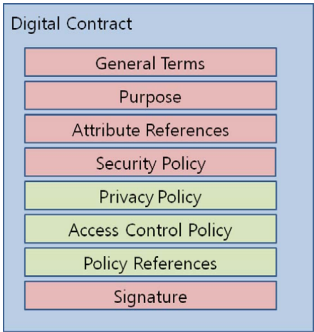


Figure 2. The Structure for Digital Contract

- (1) General Terms: the element that describes version, agreement date, validation date and any notice for a user. This is mandatory element.
- (2) Purpose: the intended use for user’s identity information. This is mandatory element.
- (3) Attribute References: This indicates which attributes this contract refers to. This is mandatory element.
- (4) Security Policy: The element can contain authentication and communication security policy that indicates how two entities can be authenticated and how communication is secured. This is mandatory element.
- (5) Access Control Policy: Any access control policy (e.g. MAC, RBAC) or authorization policy can be described in this element. This is optional element.
- (6) Privacy Policy: The element can contain any kind of privacy policy statement that is defined externally. This is optional element.
- (7) Policy References: Any policy that is defined externally can be specified here to refer to. This is optional element.
- (8) Signature: A contract can have maximum two signatures that is signed by two entities who agrees to the contract. It must be signed at least by a single entity for its validity. This is mandatory element.

C. Digital Identity Interchange Layer

Three layers are defined: Application, Identity Interchange and Network or Communication Layer. Application Layer may be a normal web or mobile computing environment. For instance, a user uses a web browser to request web service to a web server. When an entity in the application needs to request identity service or authentication, it calls the service provided in the Identity Interchange Layer. Logically the DIC is located in both application and identity interchange layer connecting the user in application layer to identity interchange layer. Every time a user tries to login to a web site, he calls the identity selector[4], which is a component of User Interface Manager in the client to select a card that represent a credential to authenticate to the web site. When a web application needs to share user’s identity information to deal with user’s service request, it can just call one of the identity interchange services that are provided in the Identity Interchange Layer.

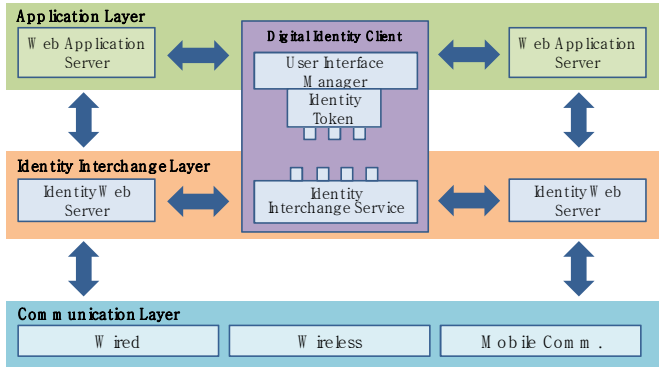


Figure 3. Identity Interchange Layer

By introducing the Identity Interchange Layer, sharing identity information between various entities can be developed and deployed independently with related to any application since an application does not need to know the detailed operation of identity interchange. Therefore identity interchange layer can provide various identity interchange related functions to existing IdM solutions in order to share identity information that can be provided by different IdM solutions.

III. GENERAL REQUIREMENTS

This section defines the following a set of requirements.

A. User requirements

DIIF(Digital Identity Interchange Framework) should satisfy the following user's requirements:

- Provides a user consistent authentication interface to support various authentication mechanisms with DIC;
- Provides a user with identity selector that he can choose what credential is going to use for authentication
- Provides a user with intuitive and consistent interface to manage his credential information with maximum security;
- Support auto fill-in registration or subscription of a web site to minimize user's interaction with the site;
- Sharing identity information should be supported any time a user wishes and he can fully control identity interchange with proper privacy protection mechanism;
- Provide a user auto updates for shared identity information when the original source is changed

Before sharing identity information, a user should have full control how security and privacy policy are established and how it is enforced to control identity interchange.

B. Functional requirements

DIIF should satisfy the following functional requirements:

- DIIF should support the integrated credential management that can manage user's credential information for authentication.
- DIIF should support identity interchange link management to provide a user with full scale view of which entities he has connections for identity interchange at present.
- DIIF should support universal authentication framework that can provide intuitive and consistent API for various authentication mechanisms.
- DIIF should support identity interchange mechanism that can provide bidirectional link to share user's identity information between entities using DIC.
- DIIF should support link contract mechanism to establish a contract for identity interchange and enforce privacy protection mechanism for the user.
- DIIF should support identity information synchronization to update the distributed and shared identity information consistently when it is changed.

DIIF should support universal token transformation to make the framework interoperable with existing identity management systems.

C. Security requirements

In order to develop secure DIIF, the following security requirements should be satisfied:

- Trust model: the security of DIIF communication will depend on the underlying trust model, which is typically based on a key management infrastructure (for example, PKI or secret key).
- Some form of transport layer security protocol should be used to provide data integrity and confidentiality when the message is in transit by a network.
- Link contract should be digitally signed by parties who reach an agreement and it could be optionally encrypted if necessary.

Data including identity information, stored in DIC should be digitally signed and encrypted when it is not in operation.

IV. DIGITAL IDENTITY INTERCHANGE FRAMEWORK

This section defines the following a set of requirements.

A. Design Principles

The framework has the following design principles to provide seamless identity interchange between entities in computing environment including mobile and ubiquitous computing.

Independent: The framework does not attach to any specific application or network environment. In another words, the framework itself should be adaptable into any environment if needed.

Pluggable: In mobile or ubiquitous computing environment, a user can work with several devices for work or entertainment. In this case what user needs is essential identity information that can establish user's identity. This information should be designed to fit into any device, and then the user can just plug his identity information into the device to use it.

Flexible: The framework should be designed flexible enough to fit into any device ranging from a workstation to a small ubiquitous computing device. The framework should be malleable enough to be configurable to adapt to various different computing environments.

Scalable: The framework itself has to be operable from a single domain to an inter-domain without introducing serious communication or computing overhead.

Interoperable: The framework should be designed in a way that operating with existing IdM system requires minimum system integration effort.

There are a few computing platform that an IdP or IdC is deployed depending on size and type of application. On the contrary, the platform that client program is installed is more diversified such as mobile phone, hand-held devices or customized proprietary devices. Therefore the client part of the framework has to be very flexible and easily configurable and pluggable to fit into any type of a device.

B. Framework Components

The following diagram below represents the framework functional components. In the figure 4, Identity Provider is the entity that manages user's identity and provides that information when the DIC or Identity Consumer requests.

Identity Consumer is the entity that requests user's identity to a DIC and provides a service using the received identity information.

The role of an entity can be an Identity Provider or Identity Consumer depending on whether the entity generates user's identity and disseminates it or requires the identity for its service. However the entity can have both roles at the same time.

DIC is the core component in this framework. DIC is the key enabler for the identity link that connects all the user's identity providers and identity consumers. A user can retrieve and updates his identity information any time it is needed using the pre-established link.

DIC consists of three parts: User Interface Manager, Identity Interchange Service and Digital Identity Token. User Interface Manager is the component that represents graphical user interface for user's identity and credential information. This component has close relationship with Web Application Server in general when a user needs to login using authentication or shares his identity information for some service. The framework only recommends what information should be managed with related to user identity and authentication credential.

Identity Interchange Service is the service part that has the responsibility for identity interchange and synchronization. Depending on what network or communication platform is used, this part should be modified to fit into the environment. For instance, the identity interchange service module in a personal computer is quite different from that of a mobile phone.

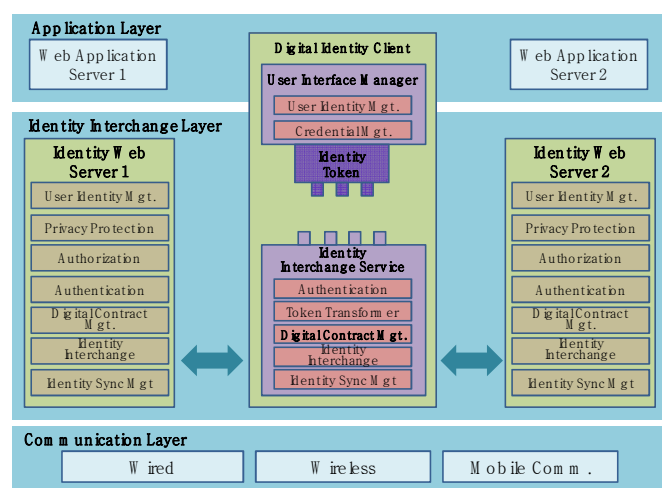


Figure 4. Digital Identity Interchange Framework

Digital Identity Token is a data model for the digital identity. It can be plugged into the identity interchange service to enable a DIC. The graphical representation of the token can be realized when it is attached with User Interface Manager. When a user switches his working environment

from a personal computer to a mobile phone, he only needs to carry the token and plug into the mobile phone. The hardware that will contain the token can be a smart card, USB token and etc.

1) User Identity Management: This is the component that manages user's identity information that will be shared by entities. In IdP, the identity management is mainly focused on a storage. On the other hand, the identity management in DIC tries to focus on graphical representation of identity information.

2) Credential Management: This is the component that manages user or site generated authentication credential information. A user can have various credentials such as password, X.509 Certificate and biometrics. The common graphical representation of credential information is defined for consistent user experience.

3) Privacy Protection: This is the component that manages privacy related function that can protect user's privacy related information. IdP collects consents from users for agreements and this information has to be managed in a way that can be easily retrieved and verified when it is necessary. This function also enforces privacy constraints, which are described in a link contract.

4) Authorization: Authorization service is to deal with making decision about user's access right and enforcing authorization decision according to user's privileges. Authorization is optional service. It is only provided when accessing to resources needs to be controlled based on user's right.

5) Authentication: This is the component that provides generic authentication framework that provides various kinds of authentication mechanisms. Authentication service includes mutual authentication for both client and servers.

6) Token Transformer: This is the component that a token issued from other IdM system is transformed into a token that can be understood and processed in DIIF. This is the gateway component to interoperate with other existing IdM system for the exchange of security tokens. This is optional service.

7) Identity Interchange: This is the core component that provides identity interchange service. Main service is to retrieve user's identity from an identity provider and to update the shared identity to an identity consumer.

8) Link Contract Manager: This is the component that manages a list of link contracts that are made between user and identity provider for authentication, access control and privacy protection. The manager manages the lifecycle of a link contract with digital signature.

9) Identity Synchronization Manager: This is the component that manages identity synchronization process. When a user changes his disseminated identity, updating the changed identity is carried out with synchronization service to all distributed IdPs.

V. CONCLUSIONS

As the federation identity IdM has many limitations when it is deployed in the Internet scale IdM, the user-centric IdM approach provides more flexible and secure solution for large scale distributed network environment. This paper proposes a framework that covers how global interoperable digital identity interchange can be achieved and how user's privacy is enhanced by providing a user more control over the process of identity interchange.

ACKNOWLEDGEMENTS

This research was supported by the IT R&D program of MKE/IITA. [2007-S-601-02, User Control Enhanced Digital Identity Wallet System].

REFERENCES

- [1] Cho, S., Cho, Y., Jin, S., *Identity Roaming Mechanism for Multiple Trust Domains using SAML v2.0 and Single Federation Bridge*, The Eleventh Annual IEEE International Symposium on Consumer Electronics 2007.
- [2] Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D., *User Centricity: A Taxonomy and Open Issues.*, In Proc. of the Second ACM Workshop on Digital Identity Management (DIM 2006) ACM Press
- [3] *Security Assertion Markup Language (SAML) V2.0*, Oasis. 2007 In <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.
- [4] *Identity Selector Interoperability Profile 1.0.*, 2007 In Microsoft <http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity-Selector-Interop-Profile-v1.pdf>.
- [5] Nadalin, A. et al. *WS-Trust 1.3*, 2007 In Oasis Web Services Secure Exchange Technical Committee, <http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html>.