Federated Authentication Mechanism with Efficient ID management

Ryu Watanabe and Toshiaki Tanaka
KDDI R&D Laboratories, Inc.
Ohara 2-1-15 Fujimino Saitama, Japan
Email: ryu@kddilabs.jp, toshi@kddilabs.jp

***Abstract*—In order to enhance user privacy and reduce management costs for identity providers, in this paper, a federated authentication mechanism with cryptographic ID management method is proposed. Based on the proposal, a proto-type was implemented and performance evaluations were carried out. The evaluation results shows feasible performance for practical implementation.**

***Keywords*-Identity management; Single Sign-on; PKI.**

I. INTRODUCTION

The OpenID [1] authentication mechanism is one of several such authentication mechanisms that make possible the realization of a single sign-on (SSO) service for Internet web sites (WEB services). Under the SSO environment, once a user is authenticated on an authentication site, the user can visit related service sites without the need for any additional authentication on each service site [2][3][4]. Therefore, the SSO technique liberates users from the nuisance of individual ID (in this paper, the term ”ID” indicates the identifier of the user account) and password handling. Therefore, users have to keep secure only one ID and password pair. The ID form in OpenID is ideally suited to existing Internet technology because it employs the URI or XRI form. Currently, OpenID is being widely adopted by Internet services such as blog sites or social network service (SNS) sites. In addition, the fact that some major service providers provide the OpenID authentication service has also contributed to its spread. Currently, however, almost all OpenID providers (OP) issue IDs to users via simple user confirmation using e-mail. Therefore, it can be said that such IDs are not assured for SSO services. For this reason, these IDs are not appropriate for economically significant services such as a shopping service. If the ID is issued via strict checking of user identity using some kind of credentials, the IDs can be regarded as having high assurance. However, this is not realistic because it is difficult to perform checks of all the IDs that have been already been issued to users through simple registration.

On the other hand, in order to make a contract for a cellular phone, users normally have to submit some kind of identification to a mobile phone company (especially in Japan). Therefore, the IDs which are associated with mobile phones are highly assured by means of the registration process. If an OpenID can acquire the assurance of a cellular phone’s ID, it can be said that the OpenID also has assurance.

For the reason given above, the authors had already proposed a federated authentication technique with cellular phone [5]. In our previous proposal, we federated OpenID with PKI-based authentication on cellular phones, which require a strong off-line identity check for contacts. By binding a user’s OpenID with an ID of his/her mobile phone, the level of assurance for user identity is increased. In addition, the mobile phone is also used in each authentication for service use. Thus, it is used like a security token and it also contribute to enhancing user authentication.

However, in our previous work, a user ID management problem remained. The OpenID basically uses a unified ID as the user identifier on each service site, called a relying party, in order to be identified as the same person on each service site. This policy is applied for user convenience. However, the use of a unified ID for a single sign-on technique causes a privacy problem called linkability. In order to resolve this problem, the use of a transient pseudonym is widely adopted. The OpenID can also use this idea. However, in this case, an OpenID provider has to handle many generated IDs. For this purpose, we introduce a cryptographically generated ID management technique. By using this technique, transient IDs are generated from a unique user ID retained by the identity provider and the identity provider only keeps the keys for ID generation. Therefore it is expected that the ID management cost imposed on identity providers can be greatly reduced. In this paper, we describe our implementation based on our proposal and also report the evaluation results.

Here, we outline the structure of this paper. In this section (Section I), the authors give the background to our research as an introduction. Then related work relevant to our research is presented in Section II. In Section III, the concept of use of cryptographical ID management is described. Then, in Section IV, an implementation based on our proposal is presented and the evaluation results are shown. Finally, our research is summarized in Section V.

II. RELATED WORK

In this section, related work relevant to our research is described. At the end of the section, the purpose of this paper is stated.

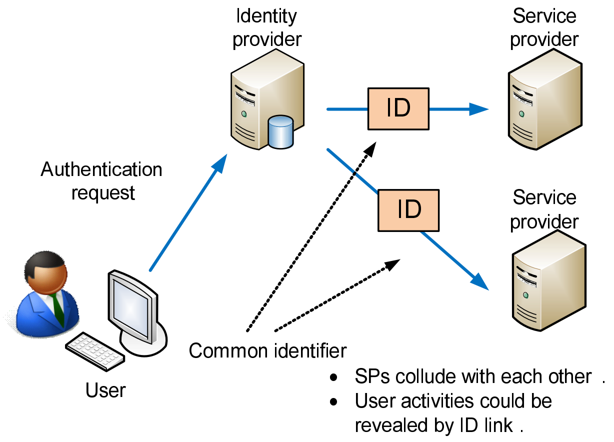

Figure 1. Problem of linkability

A. Single Sign-on (SSO)

SSO is a method of authentication and access control. Once authenticated on one identity provider, users have permission to use resources or visit various websites, etc. Therefore, this technique liberates users from password fatigue caused through having to handle many different ID and password pairs. In addition, the fact that a user merely has to keep a single unique ID password pair contributes to the enhancement of user security.

In order to realize an SSO scheme, there are many specifications or implementations. The Liberty Alliance Project (LAP. Currently, the Kantara Initiative has taken over this project.) [6][7] , security assertion markup language (SAML) [8], OpenID, and CardSpace of Microsoft are examples. Originally, they were specified or implemented only for an SSO scheme. Currently, the scope has been enlarged and they deal with user information (identity) and are referred to as an identity management mechanism (IdM).

1) Linkability: Under the SSO environment, there is a known privacy problem called ”linkability” due to the treatment of user identifiers. In the ordinary flow of an SSO technique, a service provider delegates a user authentication to an identity provider. Then, the identity provider checks the condition of local user log-on and sends the results to the service provider. In this case, in order to distinguish between the identity provider and the service provider, some identifier is used. If the same identifier is used by an identity provider and also by multiple service providers, a privacy problem arises. If service providers collude with each other, the user’s activities on each service provider are linked using this common ID (Figure 1).

B. OpenID

The OpenID mechanism is a decentralized authentication scheme for the SSO mechanism. OpenID users identify themselves with a URI and XRI. In the first specification of the OpenID mechanism [9], the idea of an identifier federation using a handle to connect an identity provider and a service provider like the SAML mechanism was not employed. In the OpenID scheme, the identity provider and service provider are referred to as the OpenID provider (OP) and relying party (RP), respectively. An RP does not have to prepare a local account (local ID) for users. Instead of a local account, an RP can use the user’s OpenID as the identifier for user identification. Under the concept of the OpenID mechanism, users are supposed to be identified with the same identity (that is a user’s OpenID) on every RP site. So the OpenID mechanism is completely unable to avoid the linkability problem because it is convenient for users to be identified as the same identity (person) on every blog site. However, the next version of the specification (OpenID Authentication 2.0 [10]) introduces an OP identifier, which indicates the OP location, to the specification and the users do not have to announce their OpenIDs to RPs. Therefore, the linkability problem can be avoided by implementation on an ID generation and distribution mechanism. Therefore, the ID generation mechanism for this problem is still an open issue. The simple answer is to use random IDs between an RP and an OP for user identification.

C. ID Assurance

The ID assurance level refers to the identity check of users; that is, when an ID provider generates a user account and issues an ID to a user, it refers to how they confirm the user’s identity.

For instance, ordinary Internet services such as blog or BBS sites require only an e-mail address for generating user accounts. A user who holds a free mail address can generate user accounts on a service. But the provider does not confirm the user’s real identity. Therefore, the provider cannot verify the existence of the user. In this case, it is said that the assurance level of the ID is low.

On the other hand, for contract of mobile phones, users usually have to show some kind of identification such as a driver’s license or passport (at least in Japan) in an off-line procedure. In this case, therefore, the assurance level of an ID bound to the mobile phone is high and the ID also provides strong assurance for online services.

The electronic authentication guideline of the NIST standard [11] (the guideline for e-authentication) is one example of a standard used for user identity proofing for registration. In the guideline, the registration levels are defined in accordance with the OMB guidance [12], which describes four identity authentication assurance levels for e-government transactions.

D. Federated Authentication Mechanism

In order to enhance the assurance level of user ID and the security level of user authentication, we have already proposed a federated authentication mechanism.

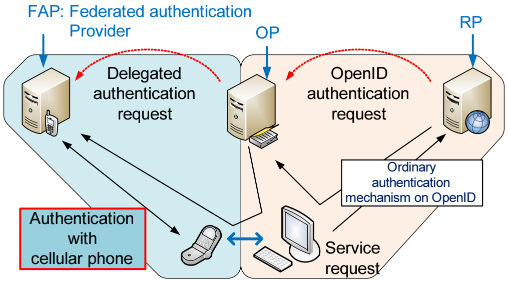
FAP: Federated authentication Provider
OP
RP
Delegated authentication request
OpenID authentication request
Ordinary authentication mechanism on OpenID
Authentication with cellular phone
Service request

Figure 2. Federated Authentication Mechanism

The concept of the mechanism is shown in Figure 2. In Figure 2, the mechanism is implemented with an OpenID authentication scheme. First, an RP delegates its user authentication to an OP. Then the OP re-delegates the request to an FAP (federated authentication provider). The FAP authenticates the user by using the user's cellular phone and returns the authentication result to the OP. The ID bound to a cellular phone has a high level of assurance. Therefore, the identity of the OpenID owner can also be confirmed. In addition, in our proposal, cellular phones are also used in each authentication. This fact also contributes to secure authentication. So cellular phones are used as a security token.

E. Objective

As described in the subsection dealing with OpenID in this section, the ID generation mechanism among an OP and RPs is not specified in an OpenID scheme and the simple answer is to generate random identifiers as handle IDs and record them bound to the user ID on an OP. However, this scheme has a problem. If the handle identifiers are persistent between an RP and an OP, the number of generated ID is limited. Therefore, it is not difficult for the OP to record them all. However, if these IDs are transient for each session even though they are used on a unique RP and OP pair in order to realize a strong anonymity for OpenID users, the number of IDs increases explosively and an enormous management cost is incurred.

III. Cryptographical ID Management

For the purpose described in previous section, we have introduced a cryptographic ID generation scheme on the OP site. By using this method, transient pseudonyms are generated from a user's unique identifier on the OP cryptographically. The binding between a user ID and the handle identifier is embedded in the handle ID itself as an encrypted form. Thus, the relationship between the user and handle identifier can be found by decrypting the handle identifier. Therefore, OPs do not have to keep all handle identifiers. Only keys for handle ID generation are stored safely at OPs. Therefore, this method contributes effectively to reducing management cost.

In our proposed cryptographic ID Management, user handle IDs are generated by means of Equation (1). Here, E_{key} means encryption with a key, and UID_{OP} means the user's ID on an OP that generates the handle ID, respectively. The value "$time$" is ID generated time and the value "$info$" is optional information for this ID generation. The mark "||" means concatenation.

$$ID = E_{key}(UID_{OP}||time||info) \tag{1}$$

The value "$time$" is used for ID type. If a user wants to use a transient ID for user identification on an RP, the ID generated time is used for this value. In this case, if the time interval is limited to a short period, the generated ID is a time-based unique ID because the value "$time$" is perfectly unique for each generated ID. In contrast, if a user wants to use a persistent value for the handle ID, a fixed value such as all zero is used for the value "$time$". In this case, the generated handle ID is always the same. Which type of ID to use depends on the policies of the user and RP. For this purpose, both users and RPs register their ID generation policy.

A. Analysis

If an IDP (OP) uses randomly generated IDs for transient pseudonyms, the number of IDs that the IDP has to keep is enormous because the IDP has to generate IDs for each user/services/sessions combination. On the other hand, by using the cryptographical ID generation method, the number of IDs managed in an IDP is greatly reduced compared to the random ID use. The relationship between a user ID on an IDP and handle ID is hidden in the generated ID itself. An IDP has to keep the encryption key for decryption. This means that the total amount of data is lessened, thereby contributing to a reduction in management cost at the IDP.

In the case of the cryptographical ID management method, the relationship among IDs is hidden in the ID itself and only the entity which knows the encryption key can decrypt generated IDs. Thus, user privacy is also protected. However, if the encryption key is leaked, the secret information could be revealed. As a measure to deal with this problem, we can use multiple keys. If different keys are used for each service, the harm is limited to a particular service. In our implementation of the prototype, therefore, dedicated keys are prepared for each RP.

IV. Implementation and Evaluation

In this section, we describe our implementation and the performance evaluation results.

A. Implementation of FAP

For the implementation, PCs (CPU: Core2Duo (E6400) 2.13GHz, memory: 1GByte) and a cellular phone are used as the FAP, OP, RP, user's PC and user's phone. For the OP and RP, a PHP based OpenID module is used. The module

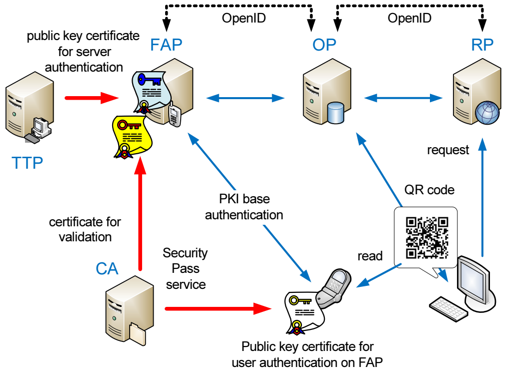
OpenID
OpenID
public key certificate for server authentication
FAP
OP
RP
TTP
request
QR code
PKI base authentication
certificate for validation
CA
Security Pass service
read
Public key certificate for user authentication on FAP

Figure 3. Implementaion overview

Table I
SYSTEM SEPCIFICATINS

Modules	
Platform	CentOS(linux 2.6.16)
OpenID module	php-opeid [13]
Crypto algorithm for ID generation	
encrypto alogorithm	AES (key length 128 bits)
Public key certificate specification on cellular phone	
Format	X.509 version 3 (RFC 2459)
Key algorithme	RSA (key length: 1024 bit)
Hash algorithm	SHA-1

is customized with a cryptographical ID generation scheme. The OP module is also customized for the communication between the OP and FAP. Between the OP and FAP, a customized OpenID module is used and ID federation can be executed between them so that they can securely exchange a handle pseudonym for their local IDs. For the authentication at the FAP, a PKI-based service (Security Pass [14]) is used. The cellular phone holds the public key certificate issued by the CA operated by a cellular phone company. The FAP also holds the public key certificate issued by a trusted third party for server authentication. Both certificates are used for mutual authentication and the construction of an SSL connection between them. In the construction of a secure connection, the FAP can extract the CN (common name) value from the certificate and use it as a user ID. Figure 3 is a schematic representation of the implementation and Table I shows the specifications for the implementation.

The flow sequence of our proposed federated authentication scheme is described below. Before this authentication flow, ID federation between an OP and FAP has been completed successfully and a handle pseudonym, which is fixed type handle ID is shared by both the OP and FAP.

1) A user accesses an RP.
2) The RP shows the login screen for the OpenID authentication mechanism.
3) The user inputs the OP identifier (in the case of OpenID authentication 2.0).
4) The RP identifies the OP from the OP identifier inputted at step (3) and redirects the user browser to the OP site.
5) The OP indicates the select screen for authentication methods (ordinary or federated authentication).
6) The user selects a method (federated authentication).
7) The OP shows the input screen for user OpenID.
8) The user inputs his or her own OpenID.
9) The OP generates an authentication delegation request to the FAP and redirects the user to FAP.
10) The user sends the authentication request to his or her cellular phone from the FAP screen.
11) The user's phone checks the URL and jumps to the FAP site on his cellular phone.
12) The FAP authenticates the user using a mobile ID and creates authentication results. Then the FAP sends the results to the OP.
13) The OP checks the results and the browser jumps to the RP site using redirection. The user can use the service on the RP.
14) The FAP sends the result to the mobile phone.

B. Implementation of ID generation

The figure shows how ID generation between OP and RP occurs. The User ID at an OP is of variable length. Therefore, we prepared a fixed length user ID bound to the user ID. In order to generate a handle ID, a fixed length ID is used. After authentication is finished using a mobile PKI service at the FAP, the authentication result is notified to the OP. At the same time, the user ID is also announced from the FAP. The ID is converted into a fixed length ID for ID generation at the OP. The procedure is described in detail below. The procedure is used for transient ID generation.

1) An OP looks up a fixed length ID (32 bits) for ID generation from the ID notified from an FAP.
2) Time value (32 bits) and reserved bits (16 bits) are added to the fixed length user ID. From the concatenated data, the CRC value is generated and also added to the data.
3) The total concatenated data is encrypted with the key for the RP which delegates user authentication to the OP.
4) The ID type identifier is added to the encrypted data. This identifier is used for ID type check at the OP and RP.
5) In addition, a service identifier is added to the data. The service identifier is assigned to each RP from the OP and used for searching for an encryption key for decrypting the handle ID at the OP.
6) The whole concatenated ID is converted to ASCII data using base 64 encoding.
7) Finally, the FQDN of the OP is added to the end of the converted data. This ID is used between the OP and RP.

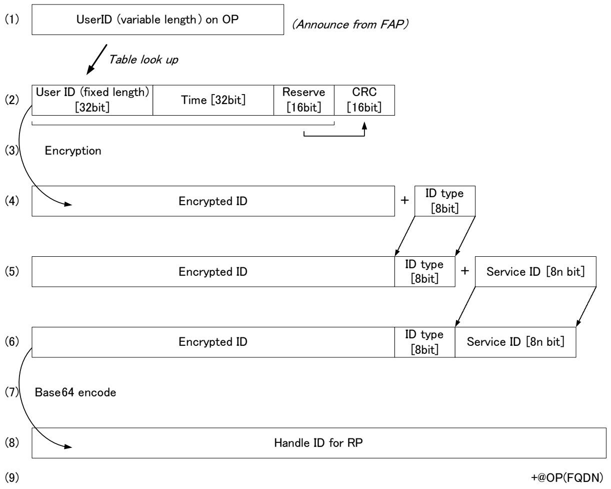

Figure 4. ID generation scheme

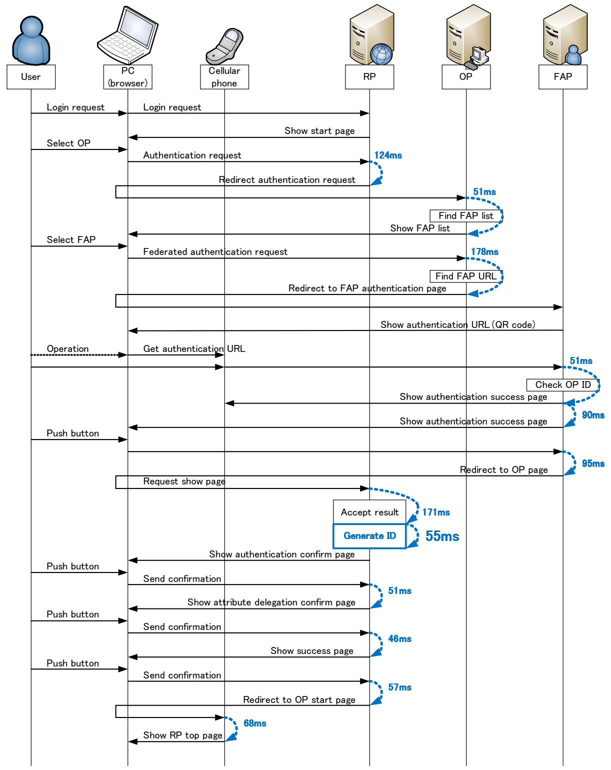

Figure 5. Federated Authentication Sequence

C. Evaluation

The time taken for user authentication on our proto-system is measured as an indication of performance. Figure 5 shows the sequence with user operation. However, the sequence for the federated authentication flow requires several manual operations such as using a camera for QR code, and pushing buttons on a cellular phone. Therefore, the total time for one operation could vary widely. Thus, we focused on the main operations on the RP, OP, and FAP. The measured time is also shown in Figure 5. The figures are the averaged value of fifteen trials. Mobile communication between the cellular phone and the FAP server depends on network conditions, and can therefore vary widely and uncontrollably. However, it was confirmed that the other operations, which do not need manual operation by users take a total of less than one second on the RP, OP, and FAP. The total time for one trial is a few seconds for ordinary cases.

In addition, the ID generation time is approximately fifty milliseconds as shown in Figure 5. This value is as fast as the table lookup case. So the total operation duration is feasible.

V. Conclusion

In this paper, we described our implementation for a PKI-based authentication scheme with cellular phone for an OpenID single sign-on scheme. For strong authentication, a public key certificate on a mobile phone is utilized; therefore, the mobile phone is used as a security token. In addition, an ID management technique based on a cryptographic technique is recommended in order to reduce management cost on the server side. Based on our proposal, a proto-system was implemented and performance evaluations were carried out. From the evaluation, each ID generation time is around 50 milliseconds. The results show that cryptographical ID generation contributes to reducing management cost in OP construction.

References

[1] OpenID, http://openid.net/ 16.08.2010.

[2] A. Pashalidis and C. J. Mitchell, “A Taxonomy of Single Sign-On Systems,” Proc. Information Security and Privacy, 8th Australasian Conference (ACIPS 2003), Springer-Verlag, 2003. vol. 2727/2003, pp. 249-264, DOI: 10.1007/3-540-45067-X_22.

[3] A. Pashalidis and C. J. Mitchell, “Impostor: A Single Sign-On System for Use from Untrusted Devices.,” Proc. IEEE Global Telecommunication Conference (GLOBECOM ’04), IEEE Press, Dec. 2004, vol. 4, pp. 2191-2195, doi:10.1109/GLOCOM.2004.1378398.

[4] T. Nishimura and H. Sato, “LESSO: Legacy Enabling SSO,” Proc. The 10th Annual International Symposium on Applications and the Internet (SAINT 2008), July 2008, pp. 301-304, doi:10.1109/SAINT.2008.76.

[5] R. Watanabe and T. Tanaka, “Federated Authentication Mechanism using Cellular Phone - Collaboration with OpenID”, Proc. 6th International Conference on Information Technology: New Generations (ITNG 2009), IEEE press, Apr. 2009, pp. 435-442, doi:10.1109/ITNG.2009.111

[6] Kantara Initiative,
http://kantarainitiative.org/ 16.08.2010.

[7] B. Pfitzmann, “Privacy in Enterprise Identity Federation - Policies for Liberty Single Signon -,” Proc. 3rd workshop on privacy enhancing technology (PET 2003), Springer-Verlag, 2003, vol. 2760/2003, pp.189-204, doi:10.1007/978-3-540-40956-4_13.

[8] OASIS SAML V2.0,
http://www.oasis-open.org/specs/index.php#samlv2.0
16.08.2010.

[9] D. Recordon and B. Fitzpatrick, “OpenID Authentication 1.1,” OpenID foundation, May 2006,
http://openid.net/specs/openid-authentication-1_1.html
16.08.2010.

[10] “OpenID Authentication 2.0 - Final,” OpenID foundation, Dec. 2007, http://openid.net/specs/openid-authentication-2_0.html 16.08.2010.

[11] “Electronic Authentication Guideline”, NIST, 2006, NIST special publication 800-63, ver. 1.0.2.

[12] “E-Authentication Guidance for Federal Agencies”, OMB, 2003, OMB M-04-04.

[13] janrain,
http://www.janrain.com/openid-enabled 16.08.2010.

[14] Security Pass, KDDI,
http://www.au.kddi.com/service/kino/securitypass/index.html
(Japanese web site) 17.08.2010.