

General Authentication Scheme in User-Centric IdM

Seung-Hyun Kim, Soo-Hyung Kim

Authentication Research Section,
Electronics and Telecommunication Research Institute,
Daejeon, Republic of Korea
ayo@etri.re.kr, lifewsky@etri.re.kr

Abstract— This paper presents a general authentication scheme for User-Centric IdM. To satisfy the authentication framework's requirements of User-Centric IdM such as consistent user experience, simple message format, and the user's participation, the proposed scheme provides a negotiation service of authentication level and a multiple authentication service in user's point of view. The proposed scheme extends the traditional general authentication framework called SASL (Simple Authentication and Security Layer).

Keywords—General Authentication, User-Centric IdM, SASL, Authentication Level, Authentication Negotiation

I. INTRODUCTION

User-Centric IdM (Identity Management) enables a user to control his/her own identity information while each site shares the user's information. Related technologies assume the authentication the site's specific method and these specifications state the authentication procedure is out-of-scope. Compared with other environments, we need to consider a user-centered authentication scheme in User-Centric IdM. The authentication scheme has to provide a consistent user experience regardless of the specific authentication mechanism. In the authentication scheme, a user can also select an authentication mechanism that meets the user's service requirement.

In survey of traditional authentication schemes, we select the SASL as an authentication mechanism in User-Centric IdM. SASL (Simple Authentication and Security Layer) is a framework for providing authentication and data security services in connection-oriented protocols via replaceable mechanisms. It provides a structured interface between protocols and mechanisms. The framework also provides a protocol for securing subsequent protocol exchanges within a data security layer. [1]-[4]

The SASL provides three kinds of service such as client authentication, client negotiation, and multiple authentications. Mechanism negotiation is protocol specific. Server advertises supported and available mechanisms to the user, and the user will then select the 'best' mechanism from this list that the user supports and finds suitable. SASL does not include the negotiation of authentication mechanism.

II. GENERAL AUTHENTICATION

A. Purposes

Consistent User Experience: While a user authenticates to a server by an authentication mechanism agreed, the user goes through consistent procedure unrelated with authentication mechanism. The proposed scheme, general authentication scheme, has single request/response protocol message because the scheme is based on the SASL. In addition, the proposed scheme has same procedure excepts only the generation / validation of security assertion which is different from each authentication mechanism. For the handling of each authentication mechanism, proposed scheme adopts plug-in model that makes easy addition and deletion. Therefore, it gives a loose to apply the proposed scheme according to the site's security policy.

Negotiation of Authentication Level: The scheme of [2], the server selects an authentication mechanism because of the server's process ability or security policy. The user does not know why the server selects the authentication mechanism. It is same when the server requires additional authentication procedure to the user. Only the SASL, the user does not know what authentication mechanism the user has to submit to acquire specific rights or services. The server only judges every decision in the traditional scheme based on the SASL.

In the User-Centric IdM, the user cannot prepare the authentication credential before the server offers its requirement of authentication mechanism. Therefore, the user has to ask constantly the list of authentication mechanisms the server provides. The server has to also inform the user extra information that helps the user to select an authentication mechanism.

Multiple Authentications: While the server provides a service to user, if the service needs unexpected rights, the user authenticates the server again with suitable level of authentication mechanism. Based on the SASL, proposed scheme provides additional authentication procedure likes the SASL. When the server requires additional authentication to the user, such as the request of higher level of service or the

expiration of authentication session, the server and the user exchange a set of general authentication request / response message. Newly established authentication session replaces previous session, and the session affects the user.

B. Sequences

Figure 1 shows a state diagram that the general authentication scheme works. Proposed scheme begins its procedures by a user asking a list of authentication mechanisms from a server. Then the server returns the list and a challenge. In case of negotiating additional authentication such as multiple authentications, previous authentication mechanisms are able to reuse. With the loaded authentication mechanisms, the user selects a specific authentication mechanism.

Next, with the user selects an authentication mechanism, the user has to select authentication medium and additional information required to use the medium. Before selection, the user can verify whether the user satisfies each mechanism that provides specific services. The user generates and sends an authentication request by using the authentication mechanism the user selects. Then the server handles the request. The server validates the request message with the authentication medium stored in the user's account and returns an authentication response. If the server sent a challenge, the server validate whether the response has a right value matched the challenge. At last, the user completes the authentication procedure according to the status message in the response.

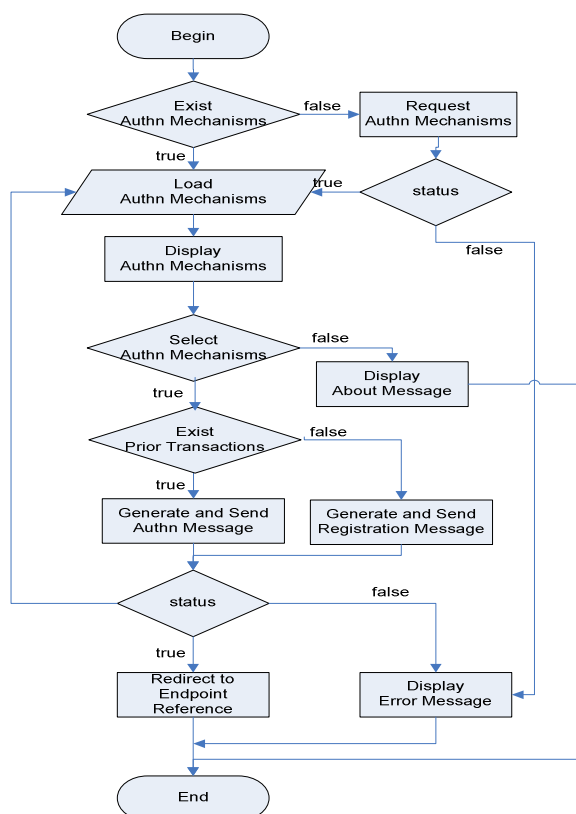


Figure 1. State Diagram of General Authentication Scheme

III. IMPLEMENTATION

We implemented the server and the client of the proposed general authentication scheme in Java version and C # version. Table 1 shows the supported types of authentication mechanisms and credentials. Description of the schemes of security token and protocol have omitted due to paper size limitation.

TABLE 1. IMPLEMENTED CREDENTIALS AND AUTHENTICATION MECHANISMS

Credentials	Authentication mechanisms
Password	cram-md5[5], plain
PKI[6]	rsa-sha1, dsa-sha1
Fingerprint	plain

Figure 2 shows a sample message of AuthMethodList. This message specifies a list of general authentication mechanisms that the server sent to the client. Based on this list, according to the authentication level configured in the pair of credentials and authentication mechanisms, the client indicates the feature that can run on the server to the user. In case of the sample in Figure 2, the server can accept plain type of password, RSA type of PKI, and plain type of biometric information. Descriptions of each authentication type are described in comment part in the message.

```

<?xml version="1.0" encoding="UTF-8" ?>
<AuthMethodList
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AuthMethod>
    <CredentialType>password</CredentialType>
    <mechanism>plain</mechanism>
    <comment>check balance </comment>
  </AuthMethod>
  <AuthMethod>
    <CredentialType>pki</CredentialType>
    <mechanism>rsa</mechanism>
    <comment>check balance, bank transfer (under 5000
dollar)</comment>
  </AuthMethod>
  <AuthMethod>
    <CredentialType>bio</CredentialType>
    <mechanism>plain</mechanism>
    <comment> check balance, bank transfer (under 10000
dollar)</comment>
  </AuthMethod>
</AuthMethodList>
  
```

Figure 2. A sample message of AuthMethodList

All request / response message in the proposed scheme is written in XML and uses the SOAP (Simple Object Access Protocol) communication. Figure 3, 4 and 5 shows sample messages of request and response. Figure 3 shows a request message to receive the list of general authentication.

The content of request message is different depending on its purpose. For example, the value of purpose element is

‘authenticate’ for user authentication as well as ‘registration’ for user registration. If a user sends an empty request with authentication purpose to a server, the server responds a list of authentication mechanisms that is registered by the user. Figure 4 shows an empty request message for authentication for the user called ‘ayo’.

```
<?xml version="1.0" encoding="UTF-8" ?>
<GeneralAuthRequest purpose="authenticate" userID="ayo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

Figure 3. A sample message of general authentication request

Figure 4 shows an example message of response from the server for a request message of Figure 3. According to this example, the status field of the message indicates that the authentication protocol for the current user should be continued. In addition, AuthMessage elements show the user’s list of registered authentication mechanisms in the server.

According to Figure 4, a user named ‘ayo’ registered a password and used the cram-md5 and plain type of authentication mechanism. Each comment field of AuthMessage element shows human-readable explanation about the feature for the verified user. A challenge field of AuthMessage element contains the challenge value for the CRAM type of password authentication or PKI authentication. In case of these types of authentication mechanisms, a request message for user authentication must include the response value for the challenge value. Finally, the EndpointReference field of the message shows the URL (Uniform Resource Locator) of server that can process the authentication request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<GeneralAuthResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Status code="continue" />
  <AuthMessage>
    <CredentialType>password</CredentialType>
    <mechanism>cram-md5</mechanism>
    <comment>You can receive a right to ...</comment>
  </AuthMessage>
  <AuthMessage>
    <CredentialType>password</CredentialType>
    <mechanism>plain</mechanism>
    <comment>You can receive a right to ...</comment>
  </AuthMessage>
  <challenge>(nonce)</challenge>
  <EndpointReference>
    (server’s soap receive url)
  </EndpointReference>
</GeneralAuthResponse>
```

Figure 4. A sample message of general authentication response for an empty request

Figure 5 shows a request message that is sent as a response to the Figure 4. According to Figure 5, the user called ‘ayo’ selected the cram-md5 type of password for user

authentication. The Data field in the AuthMessage element includes the BASE64[7]-encoded verification data for the challenge of Figure 4. The server receives the request message and verifies the verification data for the challenge with the user’s password. If the verification data is verified, the server provides its function to the user.

```
<?xml version="1.0" encoding="UTF-8" ?>
<GeneralAuthRequest purpose="authenticate" userID="ayo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AuthMessage>
    <CredentialType>password</CredentialType>
    <mechanism>cram-md5</mechanism>
    <Data>NmI5MmJmZDExZDc1YWExZTQ1NW
      NjMTI3ZTIyMzdlMjY=</Data>
  </AuthMessage>
</GeneralAuthRequest>
```

Figure 5. A sample message of general authentication request with the verification data

We develop the general authentication scheme and integrate it with the ‘Digital ID Wallet’ [8], our identity management framework. Figure 6 shows the logical architecture of the ‘Digital ID Wallet’ framework. Between the DIW (Digital ID Wallet) and the ISS (Identity Sharing Server), proposed scheme manages the authentication procedure.

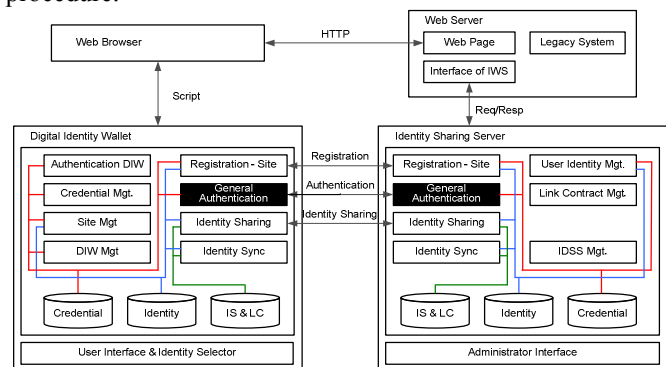


Figure 6. General Authentication Architecture

IV. ANALYSIS

Security: Proposed scheme supports message security, MITM (Man-In-The-Middle) [9] security, multi-factor authentication, and so on. Considering the service, the proposed scheme carries out a suitable authentication service. Although the proposed scheme provides multi-factor authentication, it is the easier to expose user’s authentication information the more authentication procedure. The proposed scheme exposes less authentication information because it makes a user to let a suitable authentication mechanism for the required service.

Usability: If a service needs an advanced authentication level, the user has to pass the authentication procedure again. However, the user feels annoying whenever the service the user wants to serve requires multiple authentications. In the proposed scheme, the user is able to know a proper

authentication mechanism to serve each service. In addition, the proposed scheme can eliminate the inconvenient multiple authentication because the user authenticates the service provider with a proper authentication mechanism at first.

Privacy: The login place makes the change of the authentication mechanisms and the authentication purpose. For example, in the public computer, it is safe that the user selects OTP (One-Time Password), very high and safe authentication mechanism, than the plain password. The user also establishes the list of services that the user wants to serve according to the type of the login place. Without the urgent situation, the user does not expose his high-level credentials such as the PKI certificate and biometric information. Different from the traditional schemes that do not consider the user's physical context, the proposed scheme provides the selection of authentication mechanisms with considering the user's situation

Negotiation: To make a right negotiation, the user has to know what the authentication mechanisms the service provider allows, and what the authentication mechanisms need to the services. The traditional schemes tend to require the strongest authentication mechanism regardless of the services. In case of SASL, the user seems to choose a specific authentication mechanism such as the lowest mechanisms. It limits the SASL scheme as an authentication service to choose a mechanism, not a negotiation service that considers the service the user want to serve.

V. CONCLUSIONS

In this paper, we proposed general authentication scheme in user-centric IdM. To satisfy the authentication framework's requirements of User-Centric IdM such as consistent user experience, simple message format, and the user's participation, the proposed scheme provides a negotiation service of authentication level and a multiple authentication service in user's point of view. This paper explains the general authentication scheme with the sequence and implementation. We also analyse the proposed scheme in point of security, usability, privacy, and negotiation.

ACKNOWLEDGMENT

This work was partly supported by the ICT R&D program of MSIP/IITP [B1206-15-1007, Development of Universal Authentication Platform Technology with Context-Aware Multi-Factor Authentication and Digital Signature].

REFERENCES

- [1] Melnikov, Alexey, and K. Zeilenga. "RFC 4422: Simple Authentication and Security Layer (SASL)," Network Working Group. Retrieved, pp.15-20, 2006.
- [2] Hodges, Jeff, Robert and Paul Madsen, "Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification," 2011.
- [3] Zuccherato, Robert, and Magnus Nystrom. "ISO/IEC 9798-3 Authentication SASL Mechanism," 2001.
- [4] Zeilenga, K., "Anonymous Simple Authentication and Security Layer (SASL) Mechanism," 2006.
- [5] Nerenberg, L., "The CRAM-MD5 SASL Mechanism," Work in Progress, 2008.
- [6] Solo, David, Russell Housley, and Warwick Ford., "Internet X. 509 public key infrastructure certificate and CRL profile," 1999.
- [7] Josefsson, Simon., "The base16, base32, and base64 data encodings," 2006.
- [8] Seung-Hyun Kim, Daeseon Choi, and Seunghun Jin., "Digital ID Wallet: Provide Selective Disclosure and Users Control of Identity Information," Security and Management, pp 457-462, 2009.
- [9] Asokan, Nadarajah, Valtteri Niemi, and Kaisa Nyberg., "Man-in-the-middle in tunnelled authentication protocols," Security Protocols. Springer Berlin Heidelberg, pp. 28-41, 2005.



Seung-Hyun Kim is senior researcher at Authentication Research Section, ETRI (Electronics and Telecommunications Research Institute). He received his B.S. and M.S. degree in 1998, 2002. He has great interest in authentication and privacy issues in various environments such as web and mobile.



Soo-Hyung Kim is principal researcher and team leader at Authentication Research Section, ETRI (Electronics and Telecommunications Research Institute). He received his M.S. and Ph.D. degree in 1998, 2015. He has great interest in Fintech security.