Journal of Information Processing Vol.22 No.3 472–479 (July 2014)

[DOI: 10.2197/ipsjjip.22.472]

Regular Paper

Privacy Preserving Attribute Aggregation Method without Shared Identifier Binding

Takeshi Nishimura[1,a)] Motonori Nakamura[1] Kazutsuna Yamaji[1]
Hiroyuki Sato[2] Yasuo Okabe[3]

Received: September 16, 2013, Accepted: February 14, 2014

Abstract: Identity federation is rapidly spreading, especially in the academic world. In identity federation users' credentials are stored only at their own organization, while the identity system provides authentication results and attributes to various online services, including cloud services that are hosted outside the user's organization. Attribute aggregation is a generalization of basic identity federation that allows a user to collect attributes from multiple authoritative sources. Group membership information is one of use cases, which is necessary to collaborate e.g., in an inter-organizational group. Despite the importance of privacy in identity federation, conventional methods of attribute aggregation require some identifier for a user to be shared among unrelated services, which makes correlation of user activity possible across the services. This privacy issue makes large-scale deployment of collaboration environments built on identity federation difficult. This paper proposes a new attribute aggregation method which does not require any shared identifier for services. The method has been implemented and validated as an extension of an open source federated identity software, Shibboleth. We also provide consideration about practical use of this new attribute aggregation method and comparison with existing technologies.

Keywords: privacy, attribute aggregation, identity federation, single sign-on, SAML, security

1. Introduction

Identity federation platforms that utilize Single Sign-On (SSO) technology have recently been deployed widely. In such platforms, there are Identity Providers (IdPs), which provide user authentication and attributes, and Service Providers (SPs), which provide a variety of services to users on the net. When an SP needs to identify a user, the SP sends a request for identification to the IdP the user belongs to. After the user is authenticated and attributes are retrieved, the SP receives a response containing information about the authentication from the IdP. This removes any requirement for the SP itself to directly authenticate a user. In order to provide SSO functionality beyond an organization, academic identity federations have been established in many countries [2], mainly in North America and Europe. An academic federation named GakuNin [3] was established in 2010 in Japan. In these federations, SAML 2.0 (Security Assertion Markup Language) [1] is a de facto standard protocol, and Shibboleth [4] and simpleSAMLphp [5] are the most popular implementations of IdP and SP in academic federations.

In identity federation, IdP can provide information about a user as attributes in addition to the authentication-related information. An academic federation may enumerate a list of commonly used and available attributes in its policy. An SP can request attributes about a user from an IdP and then utilize those attributes to authorize users and provide services. In general, an IdP releases attributes they derived from authoritative source systems within the organization. Although academic society affiliations or research community membership are user data that would often be interesting for academic services, a typical campus identity management system is not designed to manage such data, and the IdP is thus unable to provide it to services. Some of the advanced academic federations are beginning to make available attributes like this, especially related to the membership of a group that spans multiple organizations, by means of the virtual organization (VO) platform. There are some existing implementations which support VOs and the collaborative applications they use, such as SWITCHtoolbox [10], SURFconext [11], COmanage [12], GakuNin mAP [13], [15]. These VO platforms can be considered some of the Attribute Providers (APs) in the SAML. The SP is typically responsible for aggregation of additional attributes from APs after the initial exchange with the IdP. In the conventional APs listed above, a unique user identifier is supplied by the IdP to the SP and is shared with APs in order to look up additional attributes of that user. If the attributes managed by the AP are of interest to and thus made available to many SPs, a user's identifier is inevitably made available to many SPs.

Sharing identity information among IdPs and SPs is a key benefit of federated identity. However, as this identity information may contain private information, disclosure of information should be minimized. A user identifier can be crafted such that

[1] National Institute of Informatics (NII), Chiyoda, Tokyo 101–8430, Japan
[2] The University of Tokyo, Bunkyo, Tokyo 113–8658, Japan
[3] Kyoto University, Kyoto 606–8501, Japan
[a)] takeshi@nii.ac.jp

This paper is a revised and extended version of a paper originally presented at COMPSAC 2013 MidArch workshop.

one user has a different persistent name at each SP, which prevents correlation of users' activities. This technique is called pseudonymization and is supported by most SAML IdPs. In order to protect user privacy, current tendency is to utilize the pseudonymous identifier rather than the simple unique identifier which is shared by SPs. This approach should also be adopted in attribute aggregation process by APs.

This paper proposes a method to realize collection of user attributes from APs using a pseudonymized identifier instead of a shared unique identifier.

2. Identity and Access Management Federations

2.1 Basic Architecture of Federation

A key concept to advanced SSO is separation of authentication from authorization, allowing this work to be distributed between two types of servers in a system. One half of the functionality is provided by the IdP, which sources user information, implements authentication mechanisms, and provides the results of user authentication. The other half is provided by an SP, which provides services to users based on the result of authentication and sometimes other user attributes provided by an IdP. In any given transaction between an IdP and an SP, after the user has been authenticated at the IdP, user's information is carried from the IdP to the SP in a secure message. This message is known as an "assertion." An assertion can include user's identifier and other attributes.

In a standalone SSO system that only involves one organization, only one IdP will typically be operated. But in the case of identity federation, which spans multiple organizations, multiple IdPs will exist and there must be a mechanism for a user to choose the right IdP for his/her authentication. A Discovery Service (DS), which usually just asks users to select their home IdP, is often used in federated identity to address this challenge.

There are two ways that user attributes can be supplied from an IdP to an SP. One is called back-channel assertion exchange (**Fig. 1**), in which assertions about a user are exchanged directly between an IdP and an SP. This is called "Assertion Query" or "Attribute Query" in SAML. The other is called as front-channel assertion exchange (**Fig. 2**), in which assertions are carried over HTTP by way of user's browser using redirects (through form auto-submission) supplied to the browser through technologies such as JavaScript. Typical SAML 1.1 deployments only use back-channel assertion exchange because there was no standard way to encrypt the assertion passed in the front channel, resulting in possible disclosures of user information through e.g., browser caches or malware. Most SAML 2.0 deployments, by contrast, support both back-channel and front-channel assertion exchanges. In these systems, front-channel assertion exchanges are preferred since additional steps (6-7) may cause some problem. For example, it may be required to configure firewalls to permit direct communication between an IdP and an SP in case non-standard HTTP/HTTPS port is used.

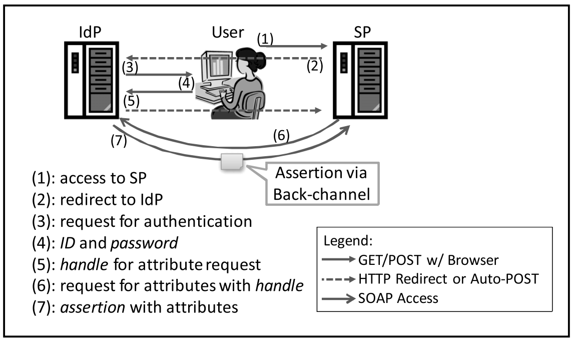

IdP
User
SP
(1)
(2)
(3)
(4)
(5)
(6)
(7)
Assertion via Back-channel
(1): access to SP
(2): redirect to IdP
(3): request for authentication
(4): *ID* and *password*
(5): *handle* for attribute request
(6): request for attributes with *handle*
(7): *assertion* with attributes
Legend:
GET/POST w/ Browser
HTTP Redirect or Auto-POST
SOAP Access

Fig. 1 Back-channel assertion exchange.

2.2 Privacy Aware Identity Disclosure

By separating authentication from authorization and attribute provision, some controlled identity disclosure methods have become widely used since identity federation works by providing personal information to outside organizations, which means user privacy should be deeply considered.

In this paper we define "privacy" as one's online activities and persona which may be personally identifying, or personally identifiable information (PII), i.e., the information is linkable directly to a specific individual. Most people want to keep privacy unless it is necessary to sacrifice it.

There are three primary types of user identifier which reveal varying amounts of information about a user: anonyms, autonyms, and pseudonyms.

Anonyms do not disclose any identity information, stating simply that this user has been successfully authenticated by the IdP. This method is helpful for access to site-licensed services such as e-journals. As these services store no PII, there are no privacy issues. But anonyms have only limited uses.

Autonyms are identifiers that are unique to users, including attributes such as eduPersonPrincipalName (ePPN), defined by MACE-Dir (Middleware Architecture Committee for Education, Directories subgroup) of Internet2 [6]. If many SPs receive these unique identifiers, they can correlate user activities through SP collusion. That is, if those SPs were to choose to merge user activities, any given SP would have a complete picture of a single user's activities across all colluding SPs. The important point is that the SP can link the activity information from another colluding SP to the specific user account stored in itself. From the point of view of privacy protection, globally unique identifiers should not be disclosed unless it is absolutely necessary for service provision [7], [8].

Pseudonyms are also unique persistent identifiers for a user, but a different pseudonym can be supplied for one user at each SP. Pseudonyms are typically calculated as a hashed value of a

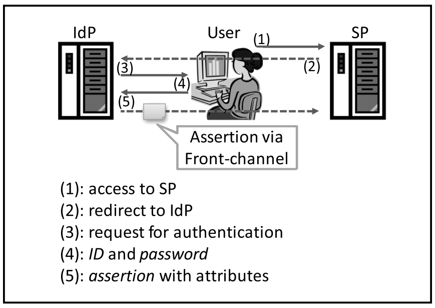

IdP
User
SP
(1)
(2)
(3)
(4)
(5)
Assertion via Front-channel
(1): access to SP
(2): redirect to IdP
(3): request for authentication
(4): *ID* and *password*
(5): *assertion* with attributes

Fig. 2 Front-channel assertion exchange.

global unique identifier and an identifier associated with the service. This type of identifier is defined as eduPersonTargetedID (ePTID) by MACE-Dir. Similar identifiers exist in other systems, such as Persistent Identifier in SAML, Private Personal Identifier (PPID) in InfoCard [8], and Pairwise Pseudonymous Identifier (PPID) in OpenID [9]. If IdPs send only pseudonyms an SP cannot link user activities from other colluding SPs to any specific user account on the SP because identities on the SP never match any identity which is contained in user activities of other SPs.

For the point of view of privacy protection, pseudonyms are suitable for most services including collaboration.

2.3 Attribute Provider

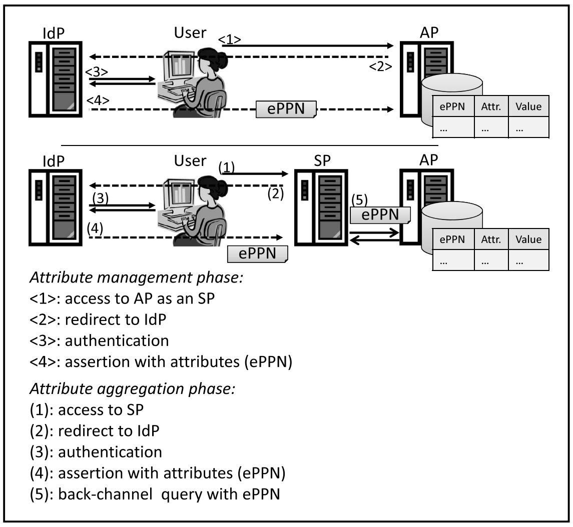
An IdP at which a user is authenticated can provide some personal information within an assertion for an SP. All the information supplied by an IdP should be guaranteed by the organization which operates the IdP and prevented from user tampering. Further, these attributes and their values should be obtained by the organization from an external authoritative data source (such as the government for name, date of birth, etc.), or originated by the organization itself (affiliation, department, e-mail, etc.). But there are always pieces of personal information managed by outside organizations other than governments. To acquire such personal information securely, utilization of APs is helpful for SPs to gather those data directly from each suitable data source. One typical application is management of membership beyond an organization. As many of these groups are not legally incorporated entities, the generic name for such groups has become "virtual organizations" (VOs).

2.4 Existing Approaches to Support Virtual Organizations

Existing VO platform such as SWITCHtoolbox, SURFconext, COmanage, and GakuNin mAP are independently developed, however, they all function similarly as an AP. For example, part of functionality of the SWITCH VO Platform, which supports SWITCHtoolbox, was implemented in Shibboleth SP version 2.2 as "simple attribute aggregation." Simple attribute aggregation aggregates attributes using back-channel assertion exchange just after user authentication.

Figure 3 shows back-channel attribute aggregation with an AP, on which simple attribute aggregation is based. In this example, both an attribute management phase and an attribute aggregation phase are illustrated. The AP acts as an SP at points in the attribute management phase. This is typical of implementation approaches for an AP providing VO membership information [15]. Users of the VO may even be permitted to manage their own membership information if allowed by administrators of counterparty IdPs or SPs, according to contractual obligations and trust. A flow of membership management in advance is expressed from ⟨1⟩ to ⟨4⟩ in Fig. 3. The expression is simplified but essentially the same as Fig. 2.

Back-channel attribute aggregation, as shown from (1) to (5) in Fig. 3, is very simple. Just step (5) is added to the sequence shown in Fig. 2.



Attribute management phase:
<1>: access to AP as an SP
<2>: redirect to IdP
<3>: authentication
<4>: assertion with attributes (ePPN)

Attribute aggregation phase:
(1): access to SP
(2): redirect to IdP
(3): authentication
(4): assertion with attributes (ePPN)
(5): back-channel query with ePPN

Fig. 3 Back-channel attribute aggregation.

2.5 Privacy Issues with Simple Attribute Aggregation

To perform back-channel attribute aggregation, a user identifier which is shared between an SP and an AP is necessary. Existing simple attribute aggregation implementations based on back-channel aggregation use autonyms, e.g., ePPN, for the user identifier. There is a privacy concern about autonyms as described in Section 2.2.

You would also use pseudonyms, e.g., ePTID, in back-channel attribute aggregation. In order to answer queries with pseudonyms from SPs, an AP must store each pseudonym for SPs in advance. As the AP collects all pseudonyms, there is also a privacy concern about collusion with AP and SPs. In any case back-channel attribute aggregation has privacy implications.

A privacy-preserving, secure attribute aggregation mechanism, which is different from back-channel attribute aggregation, will be crucial to the deployment of VOs and APs.

3. Front-Channel Attribute Aggregation

The basic concept of front-channel based attribute aggregation is to acquire assertions from an IdP and an AP in a series through the browser and deliver them to the SP. There are thus flows between the SP and the IdP, between the AP and the IdP, and between the AP and the SP. There is no requirement for an identifier shared between the SP and the AP since the IdP is the only entity performing user authentication.

Front-channel attribute aggregation is shown in **Fig. 4**. In this technique, the advance membership management phase does not require storage of ePPN. ePTID is sufficient. In order to identify the user in the attribute aggregation phase, this identifier must be persistent. The rest of the flows proceed as in Section 2.4.

Front-channel attribute aggregation requires an additional sequence of transactions, from (5) to (9) instead of step (5) of Fig. 3. This sequence makes it possible to obtain attributes associated with the user from the AP without sharing a unique identifier. This sequence also utilizes the APs ability to act as an SP. The AP will act as an SP to request user authentication of the IdP after an initial redirection from the original SP. In step (7), authentication of the user by the IdP is simplified and requiring user's cre-

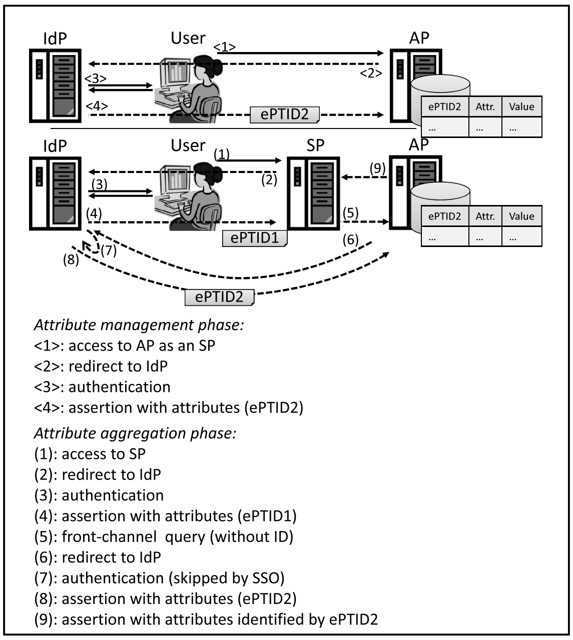

IdP
User
AP
<1>
<2>
<3>
<4>
ePTID2
ePTID2 Attr. Value
... ...
IdP
User
SP
AP
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
ePTID1
ePTID2
ePTID2 Attr. Value
... ...
Attribute management phase:
<1>: access to AP as an SP
<2>: redirect to IdP
<3>: authentication
<4>: assertion with attributes (ePTID2)
Attribute aggregation phase:
(1): access to SP
(2): redirect to IdP
(3): authentication
(4): assertion with attributes (ePTID1)
(5): front-channel query (without ID)
(6): redirect to IdP
(7): authentication (skipped by SSO)
(8): assertion with attributes (ePTID2)
(9): assertion with attributes identified by ePTID2

Fig. 4 Front-channel attribute aggregation.

dential is skipped since the user has already been authenticated to the IdP in step (3) and the session with the IdP should be still active. In the event SSO is not desired, the user may be prompted to be authenticated again.

In step (5) to (9) IdP selection by the user is also skipped. This differs from the normal authentication request process. Because the user-selected IdP is known to the SP, the SP can put the IdP's identifier, i.e., entityID in SAML term, into the request in step (5). Then the AP can utilize this information to request authentication directly.

As you may find, the IdP does not guarantee that the identity authenticated in step (2) is the same identity which is authenticated in step (7). This is because the SP and the AP share no identifier. We will describe it in more detail in Section 5.3.

4. Implementation

Our proposed front-channel based attribute aggregation mechanism was implemented within the open-source software packages distributed by the Shibboleth Project; specifically, Shibboleth SP 2.5.0 and Shibboleth IdP 2.4.0. It is designed so that IdPs are not required to redeploy the IdP with custom code to support this mechanism for front-channel attribute aggregation. This makes large-scale deployment of the proposed feature much easier. **Figure 5** shows a detailed flow sequence for front-channel based attribute aggregation, while **Table 1** shows the individual extensions that had to be added to the Shibboleth SP and IdP. As described in previous section our AP implementation consists of Shibboleth SP part and Shibboleth IdP part. The former requests user identifiers during front-channel aggregation, and it also provides user interface to manage groups and group memberships. The latter sends group membership information based on its group management database.

We also deployed it in the test instance of GakuNin mAP [13] and a mailing list service. It worked as expected. We are planning to distribute our front-channel attribute aggregation-enabled version of the Shibboleth SP for service providers to use front-channel attribute aggregation.

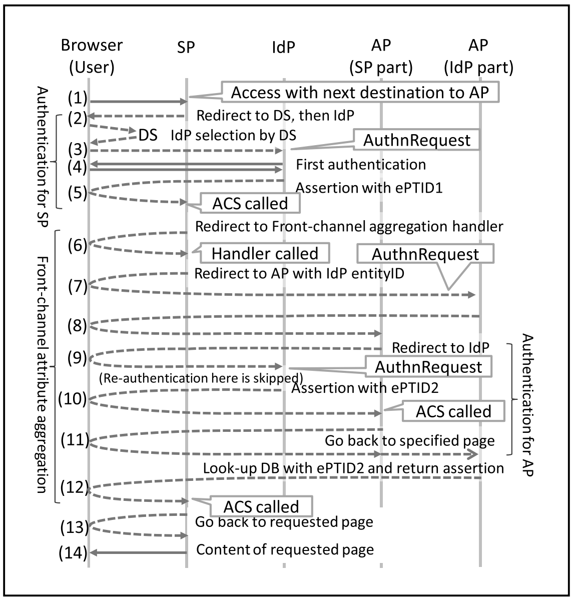


Fig. 5 Flow sequence of front-channel attribute aggregation.

Table 1 Added extentions.

IdP/SP	Added/Modified feature	Add steps
SP	AggregationSessionInitiator	837
SP	AssertionConsumerService	67
IdP (AP)	AggregationProfileHandler	243
IdP (AP)	AggregationRemoteUserLoginHandler	155

4.1 Extensions for Shibboleth SP

Our proposed method requires the SP to initiate an attribute query over the front-channel just after user authentication and aggregate the secondary received assertion from the AP with the original assertion issued by the IdP. The following changes to the SP implementation have been made to make this possible:

- A new handler, the "AggregationSessionInitiator," was implemented to initiate front-channel attribute aggregation and indicate that the user should be redirected upon successful assertion generation by the AP to an assertion consumer service endpoint URL. Most of the code is copied directly from the "SAML2SessionInitiator" in the core distribution.
- The URL associated with the "AggregationSessionInitiator" handler must be supplied as the relay state parameter (e.g., "target=") when initiating the authentication process with the IdP in step (3) in Fig. 5. This initiates user authentication and front-channel attribute aggregation in order.
- The entityID of the IdP that authenticated this user is added into a parameter in the AuthnRequest with IDPEntry element when calling an AP from the handler at step (7) in Fig. 5 as follows:

```
⟨samlp:AuthnRequest
   :
  ⟨saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:
     2.0:assertion"⟩
  https://[SP]/shibboleth-sp⟨/saml:Issuer⟩
  ⟨samlp:Extensions⟩
     ⟨gknnp:IDPEntry ProviderID="⟨IdPentityID⟩"/⟩
```

```
    ⟨/samlp:Extensions⟩
    ⟨samlp:NameIDPolicy AllowCreate="1"/⟩
  ⟨/samlp:AuthnRequest⟩
```

- The AssertionConsumerService (ACS) handler was also modified to allow it to accept the assertion from an AP and merge its values with the attributes from an IdP which already received at step (5) in Fig. 5 into a unified representation.

After all this is in place, accessing the SP with the following URL, for example, initiates the whole sequence of front-channel attribute aggregation from step (1) in Fig. 5. A login button in a page on the SP may be used to initiate the process by redirecting a browser to a URL like the following:

```
https://[SP]/Shibboleth.sso/Login?target=%2FShibboleth.sso
%2FMAP%3FentityID%3Dhttps%3A%2F%2F[AP]%2Fidp
%2Fshibboleth%26target%3D%252Fsecure
```

The Shibboleth SP part of an AP manages SAML authentication for both group management and front-channel aggregation, but each should have separated session management and sessions for front-channel aggregation should have a very short validity period. Otherwise, the front-channel method confuses the other identity included in the group management session with the same identity which the SP has received just before the front-channel aggregation.

4.2 Extensions for Shibboleth IdP to Be Used as AP

In Shibboleth, an AP may be treated as a variant of an IdP, allowing the Shibboleth IdP implementation to be extended and used as the basis for the AP for front-channel attribute aggregation.

At the IdP, a new handler called “AggregationProfileHandler” is defined to accept the aggregation attribute query from the SP in step (7) as depicted in Fig. 5. Then this handler redirects the user to the original IdP as shown in step (9) in order to get an ePTID on that user for the AP. The original IdP is specified by the SP in the AuthnRequest with an IDPEntry entityID. That is, the AP redirects the user to the following URL to get authenticated with the specified IdP using the SP functionality of the AP server. By the target parameter this URL indicates that the user should then go back to the initiating SP. The target parameter URL actually corresponds to the destination of step (11) of Fig. 5.

```
https://[AP]/Shibboleth.sso/Login?entityID=[IdP]&target=
%2Fidp%2FAuthn%2FRemoteUser
```

Another new handler, the “AggregationRemoteUserLoginHandler,” is defined in the IdP part of the AP. It processes the receipt of the response from the original IdP, as described as the destination of step (11) in Fig. 5. Then the user is finally redirected back to the SP with attributes retrieved from the AP associated with the ePTID in an assertion.

4.3 Metadata Modification to Distinguish IdP and AP

As described above, front-channel attribute aggregation flow is similar to normal authentication flow. Although the SingleSignOnService endpoint and ACS endpoint of our method should not be declared identically in metadata [1] to that of normal authentication, that is, SAML Web Browser SSO [1]. This is for policy reasons rather than technical reasons.

One important difference between these two profiles is that the authentication assertion in AP’s response is meaningless as the AP does not authenticate the user by itself. In that respect, if an AP declares such a normal authentication endpoint in its metadata, an SP may request authentication to an AP by mistake. Also, an SP receiving the assertion from an AP should merge attributes into the existing session instead of creating a new session to store attributes. If the endpoints of the SP cannot be distinguished by IdPs and APs they cannot determine where they should return assertions.

We assigned a new binding identifier for front-channel attribute aggregation. APs declare their endpoints as follows.

```
⟨md:SingleSignOnService
  Binding="urn:mace:gakunin.jp:2.0:profiles:
    FrontChannelAggregation"
  Location="https://[AP]/idp/profile/SAML2/Redirect/AP" /⟩
```

Also SPs declare their endpoints to receive assertions from APs as follows.

```
⟨md:AssertionConsumerService
  Binding="urn:mace:gakunin.jp:2.0:profiles:
    FrontChannelAggregation"
  Location="https://[SP]/Shibboleth.sso/SAML2/
    POST-FrontChannelAggregation" index="6" /⟩
```

Currently our AP implementation does not eliminate issuance of meaningless authentication assertions, but these declarations still help IdPs and SPs to avoid confusion and prevent unexpected use of endpoints.

5. Considerations

5.1 Performance

Traditional simple attribute aggregation uses back-channel query by SPs to obtain attributes from APs. Since this query is not immediately part of the login sequence performed by the user’s browser, the query does not influence access time directly. Back-channel queries also can be made concurrently in case multiple APs are used in a single session.

By contrast, front-channel queries are made with a sequence of HTTP redirects. Users will see frequent changes of URLs in the address bar of the browser, and it increases the time required to complete data transmission with the browser. In cases where the browser has a small limit on the number of consecutive redirects the user may encounter an error. In this case the sequence of redirects will not complete, attributes will not be retrieved from the AP. This limit is usually implemented to stop any infinite redirection loops. It is, moreover, not easy to reduce access time since HTTP redirects in series can’t perform concurrent access to APs.

One solution is to use an “IFRAME” or similar browser-supported technique to hide what is occurring in the front-channel from users. To request attributes to APs the SP sends the page containing a number of IFRAMEs to the browser. Each IFRAME has a URL to initiate an attribute query to each AP. This mechanism is similar to some implementations of single logout from many SPs. It performs concurrent access to multiple APs, and it can show a comprehensible description to the user when some error occurs. As this mechanism depends on a third-party cookie setting in user’s browser, we need further investigation into it.

Another option is hybrid protocol that leverages both front-channel and back-channel. When first invoked, the SP uses front-channel attribute aggregation. Within the aggregation process, the AP generates a new pairwise persistent identifier and sends it to the SP at step (12). The next time, the SP recognizes the user, and the SP can use back-channel attribute aggregation with the newly established persistent identifier. Because this hybrid protocol requires SP modification, e.g., another storage to store the persistent identifier in the SP, it should be an option when the SP require more processing speed. We mention selection of aggregation methods in Section 6.

Another performance issue is redundant redirection. In Fig. 5, there are some redundant redirections, e.g., (5), (8), (10) and (12). The initial redirects of these pairs seem to be unnecessary and could be eliminated. But these are part of the original SAML Web Browser SSO mechanism, which accepts an assertion at an AssertionConsumerService first and then redirects the user to his/her desired URL. This is called RelayState mechanism in the SAML specification. In our implementation, this mechanism is used as is to minimize extensions to the protocol and to implementations. Only step (8) is not related to RelayState mechanism and even this can be eliminated if we tightly integrate a Shibboleth SP with a Shibboleth IdP.

5.2 Same Attributes from Multiple APs

The IdP and APs may provide different attribute values associated with the same attribute name. In such situations, these values are concatenated with a semi-colon into a single attribute value in Shibboleth SP. But some services may want to know which authority asserted each value. This is an issue not only for front-channel attribute aggregation, but for back-channel attribute aggregation as well. Some sort of new mechanism to distinguish the source of any given attribute is needed generally.

5.3 Security

Front-channel attribute aggregation is processed by the user's browser as shown in Fig. 5. The user can interrupt this sequence by prohibiting storage of their selection of IdP with a DS, or by clearing session cookies at each AP. The user can also switch to another IdP for authentication that is associated with different APs, which can provide different attributes. This type of abuse can be avoided by the SP comparing entityIDs in the assertions issued by and returned from the IdP and APs.

Another issue is that the same IdP may authenticate the user as another user at a different time in case he/she has multiple accounts on the IdP. For example, a user can be authenticated as User1 for an SP and User2 for an AP. In most cases, every user only has an account (identifier), and this will not be an issue. But some organizations issue multiple accounts for a single user with multiple different roles within the single organization. Some SPs may create accounts on them which reflect these roles and attributes provided by APs also reflect these roles. For example, assume an SP has Account1 and Account2 which reflect Role1 and Role2 correspondingly. If Account2 does the operation which only Role1 is permitted, the SP with strict policy would consider it as access violation. This issue can be caused if the operation is partly authorized based on front-channel aggregated attributes from the AP.

There is no such issue in a typical VO environment, which means that SPs authorize users based on only attributes from one AP. Care must be taken when there are multiple APs (including IdP) and users are authorized based on attributes from different sources. It is one of the choices to use autonyms like ePPN and back-channel attribute aggregation for such complicated but important SP.

6. Related Work

If you want to deploy non-SAML based attribute aggregation method in your federation, e.g., OpenID Connect [14], all existing IdPs must be modified in order to support OpenID, which may not be feasible in any existing federations. But we still think it is important to compare the functionality of our implementation in SAML with other possibilities in other protocols.

There are two types of attribute aggregation methods defined in the OpenID Connect specification [14]. One method is known as an aggregated claim, while the other is known as a distributed claim. In an aggregated claim, an IdP (referred to as an OP in OpenID) collects attribute information directly from APs, so this type of claim is only suitable for aggregation use cases that don't require privacy. In a distributed claim, an IdP collects access tokens as keys to get attributes from APs and sends them all in a bundle to an SP (referred to as an RP in OpenID). Although access tokens are not globally unique identifiers, users must still place complete, universal trust in their IdP, since that IdP has the capability to know all linked information.

Users of our front-channel aggregation can confirm that the attribute exchange can only be performed with their involvement. SPs cannot request independently as they have no shared identifier. This property is expected by people who are familiar with SAML. On the other hand, the distributed claim method in OpenID Connect assumes that SPs will request claims for some time after user authentication though it depends on the validity period of each access tokens.

Table 2 presents classification of attribute aggregation methods sorted by the type of user identifier which is necessary for SPs so that the method works correctly, including both user identifiers which SPs must obtain in advance or SPs know eventually. The aggregated claim of OpenID Connect is not considered because it is not suitable for situations where privacy is essential, as described above.

The columns of Table 2 are closely related to the resulting level of privacy. For example, a distributed claim in OpenID Connect is more privacy-preserving than back-channel queries in SAML (e.g., simple attribute aggregation in Shibboleth) because the latter cannot prevent SPs from correlating user activity across

Table 2 Classification of aggregation methods.

User identifier obtained by SP	Anonym	Pseudonym e.g. ePTID/PPID	Autonym e.g. ePPN
SAML-based methods	front-channel	front-channel hybrid	back-channel
OpenID-based methods	-	distributed claim	-

SPs. We classified the distributed claim of OpenID Connect as pseudonymous because an IdP sends access tokens to an SP and the access token is tend to be reused multiple times during its validity period. Moreover, in some implementations, the access token contains a PPID for the sake of efficiency.

In contrast, our method (front-channel aggregation of SAML) does not need any specific identifier to be issued to any given SP. If a distributed claim implementation were to never reuse access tokens and obtain a new access token every time, it is equivalent to our front-channel method. We don't think this is a usual case when using OpenID Connect.

An example service which would require only an anonym is an e-Journal service giving access permission to some small groups [15]. We think our anonym-based method is preferred by such services which have no direct need for user identifiers and want to do access control based on group membership.

As described in Section 5.3, the use of anonyms results in some uncertainty about user identity. In some cases of front-channel attribute aggregation, a user can trick an SP into believing an ID has attributes of another ID from the same IdP, and it is difficult for the SP and the AP to detect. We think it is important for an AP to provide several aggregation methods which correspond to the required level of privacy, and for each SP to select a proper method in accordance with the essential set of attributes expected by the SP and the processing speed required. The aggregation method could be automatically selected based on the type of identifier that the SP has for the user at the moment of initiating attribute aggregation.

In the aggregated claim method of OpenID Connect theoretically one can encrypt user attributes in an AP in order to decrypt them in an SP. Because an IdP cannot decrypt the encrypted attributes when they pass through the IdP, there is no impact on user privacy. Likewise, in back-channel aggregation of SAML, one can encrypt the user identifier at the time of issuance by an IdP in order to decrypt it in an intended AP. Because an SP cannot decrypt the encrypted identifier when it passes through the SP, there are no privacy implications. Though there are specifications about encryption, an additional specification would be necessary to pass the final destination from a requesting entity to an entity which will do encryption. To the best of our knowledge there is no such specification and also there is no such implementation.

The Danish academic identity federation WAYF is operating an attribute collector known as JAKOB [18]. JAKOB aggregates attributes and sends them to the IdP, similar to OpenID's aggregated claim aggregation. Certainly it does not send any identifier. But it needs modification of the IdP so it is feasible if there are only one or some IdPs in the federation. Also it is not preferred in some cases that the IdP knows all passed information. We also believe that SP-initiated flow is natural because the SPs know what attributes they need.

Attribute aggregation is being widely developed not only in academic federations but also commercial and governmental identity federations, such as Backend Attribute Exchange (BAE) v2 [17]. Though BAE is based on back-channel attribute exchange, our method can be combined with such technologies in this sort of environment where there are multiple APs. It is rather easy to combine our implementation with other technology, as our implementation defines an independent handler to initiate front-channel attribute aggregation to one AP. In contrast, the simple attribute aggregation method in Shibboleth is tightly bound with user authentication and cannot be initiated independently.

Inman et al. [16] proposes an attribute aggregation method among IdPs in which the same user has multiple accounts. They link the accounts and exchange the identifiers in advance. The scope of our system does not cover aggregation of attributes from multiple IdPs, but it can deal with multiple accounts of the same user through account linking by storing multiple ePTIDs from different IdPs in one account on the AP.

Solberg [19] suggests modifying the SAML AttributeQuery protocol to support front-channel. Though it is semantically correct except for the semantics of the ⟨Subject⟩ element in the query, we selected Web Browser SSO protocol as the basis for our work because it is relatively well defined for use in the front channel and it minimizes modification points of existing systems.

7. Concluding Remarks

Privacy must be deeply considered to encourage widespread deployment of identity federation and moreover user collaboration in VOs. This collaboration on identity federation requires distributed APs which provide user's attributes instead of an IdP. This paper presents a front-channel based attribute aggregation method to gather and utilize attributes provided by APs without the requirement that SPs possess a globally unique identifier for the user. Our method helps to avoid correlation of user's activities across SPs. We implemented the proposed method as an extension of an existing system. This implementation confirms that the proposed method works properly and a limited set of modifications is needed, which is preferable for easy and wide-scale deployment.

Acknowledgments This work has been funded by grants from Japan's Ministry of Internal Affairs and Communications (Strategic International Cooperation R&D Promotion Program in FY2012: Privacy Enhancement for Open Federated Identity/Access Management Platforms).

References

[1] Cantor, S., Kemp, J., Philpott, R. and Maler, E. (Eds.): Security Assertion Markup Language (SAML) V2.0 (Mar. 2005), available from ⟨http://saml.xml.org/saml-specifications⟩.

[2] REFEDS (Research and Education Federations): REFEDS Federation Survey, available from ⟨https://refeds.terena.org/index.php/Federations⟩ (accessed 2013-09-16).

[3] GakuNin: Academic Access Management Federation in Japan, available from ⟨https://www.gakunin.jp/⟩ (accessed 2013-09-16).

[4] Shibboleth Consortium, available from ⟨http://shibboleth.net/⟩ (accessed 2013-09-16).

[5] SimpleSAMLphp, available from ⟨http://simplesamlphp.org/⟩ (accessed 2013-09-16).

[6] Internet2: eduPerson & eduOrg Object Classes, available from ⟨http://middleware.internet2.edu/eduperson/⟩ (accessed 2013-09-16).

[7] Pimenta, F., Teixeira, C. and Pinto, J.S.: GlobaliD: Federated identity provider associated with national citizen's card, *2010 5th Iberian Conference on Information Systems and Technologies* (*CISTI*), pp.1–6 (2010).

[8] Nanda, A.: Identity Selector Interoperability Profile V1.0 (2007), available from ⟨http://www.microsoft.com/en-us/download/details.aspx?id=18221⟩.

[9] Federal Identity, Credentialing, and Access Management: OpenID 2.0 Profile (2009), available from ⟨http://www.idmanagement.gov/sites/default/files/documents/ICAM_OpenID20Profile.pdf⟩.

[10] SWITCH: SWITCHtoolbox, available from ⟨http://www.switch.ch/toolbox/⟩ (accessed 2013-09-16).

[11] SURFnet: SURFconext, available from ⟨http://www.surfnet.nl/en/Thema/coin/Pages/Default.aspx⟩ (accessed 2013-09-16).

[12] Internet2: COmanage: Collaborative Organization Management, available from ⟨http://www.internet2.edu/comanage/⟩ (accessed 2013-09-16).

[13] GakuNin mAP, available from ⟨https://map.gakunin.nii.ac.jp/map/⟩ (accessed 2013-09-13).

[14] Sakimura, N., Bradley, J., Jones, M., de Modeiros, B., Mortimore, C. and Jay, E.: OpenID Connect Messages 1.0 – draft 20 (2013), available from ⟨http://openid.net/specs/openid-connect-messages-1_0.html⟩.

[15] Nishimura, T., Nakamura, M., Otani, M., Yamaji, K. and Sonehara, N.: Group Management System for Federated Identities with Flow Control of Membership Information by Subjects, *Proc. 2012 IEEE 36th International Conference on Computer Software and Applications Workshops* (*6th IEEE International Workshop on Middleware Architecture in the Internet* (*MidArch2012*)), pp.94–99 (2012).

[16] Inman, G. and Chadwick, D.: A privacy preserving attribute aggregation model for federated identity managements systems, Upgrade, Vol.11, No.1, pp.21–26 (2010).

[17] Backend Attribute Exchange (BAE) v2.0 Overview (2010), available from ⟨http://www.idmanagement.gov/sites/default/files/documents/BAE_v2_Overview_Document_Final_v1.0.0.pdf⟩.

[18] JAKOB, available from ⟨http://wayf.dk/en/component/content/article/497⟩ (accessed 2013-09-16).

[19] Solberg, A.: SP-Centric Attribute Aggregation, available from ⟨https://rnd.feide.no/2009/08/24/sp-centric_attribute_aggregation/⟩ (accessed 2013-09-16).

Takeshi Nishimura graduated from the University of Tokyo, Japan, where he received B.Sc. and M.Sc. degrees in information science in 1996 and 1998, respectively. From 2001, he was a research associate at the University of Tokyo. Since 2009, he is a project researcher at National Institute of Informatics, Japan. His research interests are authentication and authorization for federated identity, identity federation management and public key infrastructure (PKI).

Motonori Nakamura graduated from Kyoto University, Japan, where he received B.E., M.E. and Ph.D. degrees in engineering in 1989, 1991 and 1996, respectively. From 1994, he was an assistant professor at Ritsumeikan University. From 1995, he was an associate professor at Kyoto University. Currently he is a professor at National Institute of Informatics, Japan (NII). His research interests are message transport network systems, network communications, next generation internet and Identity & Access Management. He is also a member of IEEE, ISOC, IEICE, IPSJ and JSSST.

Kazutsuna Yamaji received Ph.D. in Systems and Information Engineering from Toyohashi University of Technology in 2000. Currently he is an associate professor at the National Institute of Informatics, Japan. His research interests include open science, data sharing and identity federation. He is a member of IPSJ, IEICE and JSIK.

Hiroyuki Sato is an associate professor at the University of Tokyo. He received B.Sc., M.Sc. and Ph.D. degrees from the University of Tokyo in 1985, 1987, 1990, respectively. Majoring: Computer Science and Information Security.

Yasuo Okabe received M.E. from Department of Information Science, Kyoto University in 1988. From 1988 he was an instructor of Faculty of Engineering, from 1994 he was an associate professor of Data Processing Center, and from 1998 he was an associate professor of Graduate School of Informatics, Kyoto University. He is now a professor of Academic Center for Computing and Media Studies, Kyoto University. Ph.D. in Engineering. His current research interest includes Internet architecture, network security and distributed algorithms. He is a fellow of IEICE and a member of IPSJ, ISCIE, JSSST, IEEE, and ACM.