2013 IEEE 37th Annual Computer Software and Applications Conference Workshops

Privacy Preserved Attribute Aggregation to Avoid Correlation of User Activities Across Shibboleth SPs

Motonori NAKAMURA, Takeshi NISHIMURA, Kazutsuna YAMAJI
National Institute of Informatics (NII)
Tokyo, Japan
{motonori, takeshi, yamaji}@nii.ac.jp

Hiroyuki SATO
The University of Tokyo
Tokyo, Japan
schuko@satolab.itc.u-tokyo.ac.jp

Yasuo OKABE
Kyoto University
Kyoto, Japan
okabe@i.kyoto-u.ac.jp

***Abstract*—Privacy is one of the most important issues in Identity Federation, a technology in which local IDs and credentials such as passwords managed at one site may be used to access many online services, including cloud services provided outside of users' organization. Attribute aggregation is an advanced technique that may be employed in identity federation, collecting attributes about a user from multiple distinct identities to provide a complete picture about a user necessary for some services. However, conventional methods of attribute aggregation require a persistent shared unique ID. This may restrict the use of federated identity for some services because these unique ID's could be used by bad actors to correlate user activity or user data. This paper proposes a new method of attribute aggregation that doesn't require a universal unique ID. SAML, a widely used federated identity standard, is used as the basis for this work. This privacy-preserving attribute aggregation technique has been validated with a successful implementation for the open source federated identity software project Shibboleth.**

Keywords-privacy; authenticatoin; federation; single sign-on

I. 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 authentication and attributes, and Service Providers (SPs) which provide a variety of services 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 and receives a response containing information about the authentication from the IdP instead of having an authentication mechanism within the SP itself.

Sharing identity information on users among IdPs and SPs is a key benefit of federated identity. But this identity information may contain private information, so disclosure of information should be minimized. A user identifier can be crafted such that 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 platforms using SAML (Security Assertion Markup Language) [1] and OpenID [2].

IdPs can provide additional information about users as attributes. But a user has many attributes, possibly each with different authoritative sources — affiliation with an organization, registered academic societies, medical records, and so on. Such additional information should be provided by Attribute Providers (APs) that are authoritative for the information and can thus ensure the quality of the provided attributes. Simple AP implementations require a unique identifier that is shared with the initial IdP to collect additional attributes about a user. If such APs may be accessed from many SPs, SPs should share user's identifier.

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

II. IDENTITY AND ACCESS MANAGEMENT FEDERATIONS

A. Single Sign-On Technologies

There are a variety of online services even within one organization, such as a university, research institute, or company. Many services require that the user be identified since most services are personalized and resources and data associated with the user are maintained.

In the early stages of identity management deployment, a username and password were provided to each user by each service to login, requiring users to remember different login information for every service. To improve on this situation, later implementations shared a username and password for many services by providing a unified authentication database such as LDAP[3]. In this case, each service still authenticates the user directly, and users are required to enter a username and password at the beginning of access to each service.

An evolutional technology called single sign-on(SSO) was then introduced to optimize authentication. Once a user provides a username and password to access one service, other services that share the same SSO system can also be used without re-entering the same authentication information. This separates authentication from service provision and unifies that authentication across services.

There are many SSO systems. CAS [4], OpenAM [5] and JOSSO[6] are some example open source SSO implementations. Most of them use proprietary protocols for SSO and are used only internally in an organization.

978-0-7695-4987-3/13 $26.00 © 2013 IEEE
DOI 10.1109/COMPSACW.2013.52

Recently, SaaS (Software as a Service) and cloud services has become widespread. Supporting these services requires SSO technologies that can be used as an authentication mechanism beyond an organization. SAML and OpenID are internationally recognized standard protocols for inter-organizational SSO widely deployed for authentication for services offered on the Internet. Authentication through SSO technologies that extend beyond an organization is referred to as federated identity.

In order to provide a trust infrastructure for federated identity, academic identity federations have been established in many countries[7] mainly in North America and Europe. An academic federation named GakuNin[8] was established in 2010 in Japan. SAML is a de facto standard in academic field and Shibboleth[9] and SimpleSAMLphp[10] are most popular implementations for academic federations.

B. *Basic Archtecture 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 performed by the IdP, which manages user information and provides the results of user authentication to SPs. The other half is 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, after the user has authenticated at the IdP, that user's information, including identity and other attributes, is carried by the user to the SP from the IdP in a secure message known as an "assertion."

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 that user's 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(a)), in which assertions about a user are exchanged directly between an IdP and an SP. The other is called as front-channel assertion exchange (Fig. 1(b)), 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 encryption of 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 it does not require configuration of firewalls to permit direct communication between an IdP and an SP.

C. *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. There are 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. 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[11]. These unique identifiers may be received by many SPs, allowing correlation of user activities through SP collusion. From the point of view of privacy protection, globally unique identifiers should not be disclosed unless it absolutely required for service provision[12][13]. 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 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 persistentId in SAML, Private Personal Identifier (PPID) in InfoCard[13], and Pairwise Pseudonymous Identifier (PPID) in OpenID[14].

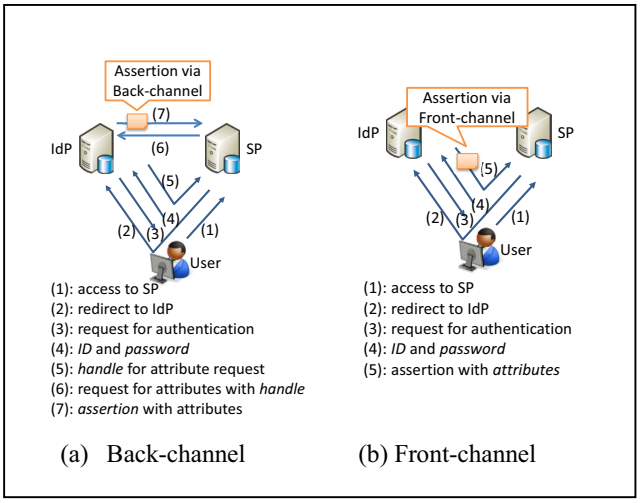

Figure 1. Back-channel and Front-channel Assertion Exchanges

D. *Attribute Providers*

An IdP at which a user authenticates 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 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).

E. Existing Approaches to support Virtual Organizations

There is a variety of tools which support collaborative work on the Internet. Most of these collaborative tools require information about users. Users may want to store group membership information about themselves and other members. It is hard to utilize if the information is stored by a tool site itself, and keeping data remote to applications is especially important to avoid provider lock-in. There are some existing implementations which support VOs and the collaborative applications they use, such as SWITCH toolbox[15], SURFconext[16], COmanage[17], GakuNin mAP[18][19]. Part of functionality of the SWITCH VO Platform, which supports SWITCHtoolbox, was implemented in Shibboleth 2.2 as "simple attribute aggregation."

F. Privacy Issues with Simple Attribute Aggregation

VO's can act as exemplary APs in identity federations. The simplest implementations of VO as AP, like other simple aggregation implementations, require a shared unique identifier. The only use of this identifier in these implementations is to obtain additional personal information including group membership.

An AP framework should be designed for easy deployment since group aware collaborative tools may be new killer applications, deployers of VOs often don't have lots of resources for identity management, and utilization of pseudonymous identifiers is one of the most important features offered by identity federation. No currently implemented AP supports the use of pseudonyms to acquire attributes.

There are two types of attribute aggregation methods defined in the OpenID Connect specification [20]. One is 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 global unique identifiers, a user must still place complete, universal trust in their IdP, since that IdP has the capability to know all linked information.

A privacy-preserving, secure attribute aggregation mechanism will be crucial to the widespread deployment of APs.

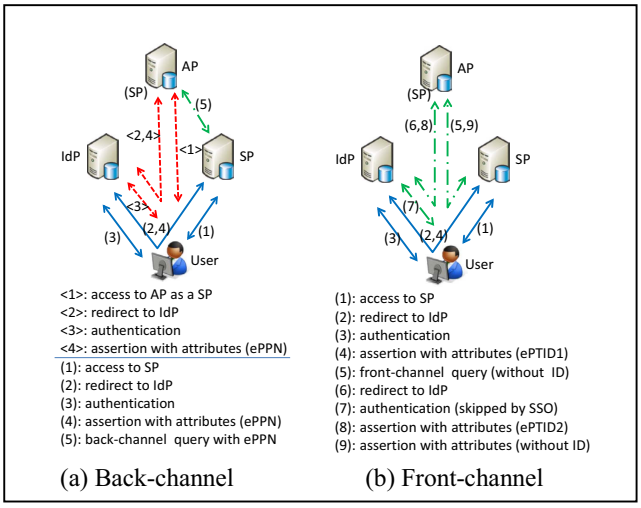

Figure 2. Back-channel and Front-channel Attribute Aggregations

III. 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.

Fig. 2 shows two types of attribute aggregation with an AP. In this example, the AP acts as an SP at points in the flows. This is typical of an implementation approach for an AP providing VO membership information. Users of the VO may even be permitted to manage their own membership information if allowed by administrators of a counter-party IdP or SP, according to contractual obligations and trust. A flow for this type of membership management is expressed from <1> to <4> in Fig. 2 (a). The expression is simplified but essentially the same as Fig. 1 (a). The same access should also be shown in Fig. 2 (b), but it was omitted here to simplify Fig. 2 (b).

Back-channel attribute aggregation, as shown from (1) to (5) in Fig. 2 (a), is very simple. Just step (5) is added to the sequence shown in Fig. 1. On the other hand, front-channel attribute aggregation in Fig. 2 (b) requires an additional sequence of transactions, from (5) to (9) instead of step (5) of Fig. 2 (a). 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 by the IdP just after an initial redirection from the original SP. In step (7), authentication of the user by the IdP is skipped since the user already

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 authenticate again.

IV. 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. Fig. 3 shows a detailed flow sequence for front-channel based attribute aggregation, while table I shows the individual extensions that had to be added to the Shibboleth SP and IdP.

A. *Extensions for Shibboleth SP*

Our proposed method requires the SP to initiate an attribute query over the front-channel 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 perform 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 "AggregationSession Initiator" handler must be supplied as the relay state parameter (e.g. "target=") when initiating the authentication process with the IdP in step (2) in Fig. 3.
- The entityID of the IdP that authenticated this user is added into a parameter in the AuthnRequest with IDPList element when calling an AP from the handler at step (8) in Fig. 3 as follows:

```
<samlp:AuthnRequest
      :
  AssertionConsumerServiceURL="https://[SP]/Shibboleth.sso/SAML2/POST"
  Destination="https://[AP]/idp/profile/SAML2/Redirect/SSO"
      :
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    https://[SP]/shibboleth-sp</saml:Issuer>
  <Scoping>
    <IDPList>
      <IDPEntry ProviderID="<IdPentityID>"/>
    </IDPList>
  </Scoping>
  <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 (17) in Fig. 3 into a unified representation.

After all this is in place, accessing the SP with following URL, for example, initiates the whole sequence of front-channel attribute aggregation from step (1) in Fig. 3. 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=/Shibboleth.sso/MAP%3FentityID%3Dhttps%3A%2F%2F[*AP*]%2Fidp%2Fshibboleth%26target%3D/secure

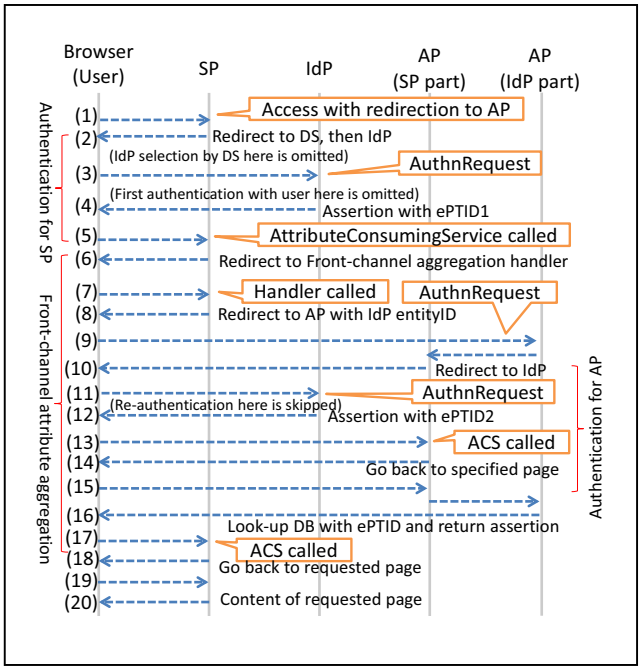

Figure 3. Flow Sequence of Front-channel Attribute Aggregation

B. *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.

A new handler at the IdP, the "AggregationProfile Handler", is defined to accept the aggregation attribute query from the SP in step (9) as depicted in Fig. 3 and redirect the user first to the IdP specified in the AuthnRequest with an IDPEntry entityID by the SP for authentication of the user to get an ePTID on that user for the AP. Thus, the AP redirects the user to the following URL to get authenticated with the specified IdP using the SP functionality of the AP server and indicate that the user should then go back to the initiating SP:

https://[*AP*]/Shibboleth.sso/Login?entityID=[*IdP*]&target=/idp/Authn /RemoteUser

Another new handler, the "AggregationRemoteUser Login Handler", is defined in the AP to process the receipt of the response from the IdP, as shown in step (15) of Fig. 3, the user is finally redirected back to the SP with attributes retrieved from the AP associated with the ePTID in an assertion.

TABLE I. ADDED EXTENSIONS

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

V. CONSIDERATIONS

A. 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 limit on the number of redirects that may be performed in series, usually implemented to stop any infinite redirection loops, the sequence of redirects will not complete, attributes will not be retrieved from the AP, and the user will encounter an error. It is, moreover, not easy to reduce access time since HTTP redirects can't perform concurrent access to APs.

One solution is to use an "IFRAME" or something feature to hide what is occurring in the front-channel from users.

Another issue on performance is redundant redirection. In Fig. 3, there are some redundant redirections, e. g. (5, 6) - (7, 8), (13, 14) - (15, 16) and (17, 18) - (19, 20). Leading redirections of these pairs seems to be eliminated. But these are in tradition of original Shibboleth SSO mechanism which accepts assertion with AssertionConsumerService first and then redirects to objective URL. In our implementation, this mechanism is used as is to minimize extensions.

B. 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. 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 origination is needed generally.

C. Security

Front-channel attribute aggregation is processed by the user's browser as shown in as Fig. 2 (b). 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 to authenticate as another user with the same IdP in case the user have multiple accounts on the IdP. For example, a user can authenticate as User1 for an SP and User2 for an AP. There might be confusion if this type of situation occurs. In most cases, every user only has an account (identifier), and this will not be an issue. But some organization issues multiple accounts for a user who has different roles in the same time.

VI. CONCLUSION

Privacy must be deeply considered to encourage widespread deployment of identity federation. This paper presents a front-channel based attribute aggregation method to gather and utilize a user's attributes provided by distributed APs without requiring a global unique identifier, which helps to avoid correlation of user's activities. The proposed solution was implemented using Shibboleth/SAML as a platform, which confirmed that the proposed method works properly.

ACKNOWLEDGMENT

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] S. Cantor, J. Kemp, R. Philpott, and E. Maler ed., "Security Assertion Markup Language (SAML) V2.0," http://saml.xml.org/saml-specifications, March 2005.

[2] OpenID Foundation, "OpenID Foundation website," http://openid.net/, last visited Apr. 1, 2013.

[3] M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)," The Internet Society, RFC2251, 1997.

[4] Central Authentication Service Project, http://www.jasig.org/cas, last visited Apr. 1, 2013.

[5] OpenAM, http://forgerock.com/what-we-offer/open-identity-stack/openam/, last visited Apr. 1, 2013.

[6] Java Open Single Sign-On Project, http://www.josso.org/, last visited Apr. 1, 2013.

[7] REFEDS (Research and Education Federations), "REFEDS Federation Survey", https://refeds.terena.org/index.php/Federations, last visited Apr. 1, 2013.

[8] GakuNin: Academic Access Management Federation in Japan, https://www.gakunin.jp/, last visited Apr. 1, 2013.

[9] Shibboleth Consortium, http://shibboleth.net/, last visited Apr. 1, 2013.

[10] SimpleSAMLphp, http://simplesamlphp.org/, last visited Apr. 1, 2013.

[11] Internet2, "eduPerson & eduOrg Object Classes", http://middleware.internet2.edu/eduperson/, last visited Apr. 1, 2013.

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

[13] Arun Nanda, "Identity Selector Interoperability Profile V1.0," http://www.microsoft.com/en-us/download/details.aspx?id=18221 . 2007.

[14] Federal Identity, Credentialing, and Access Management, "OpenID 2.0 Profile," http://www.idmanagement.gov/documents/ICAM_OpenID20Profile.pdf, 2009.

[15] SWITCH, "SWITCHtoolbox," http://www.switch.ch/toolbox/, last visited Apr. 1, 2013.

[16] SURFnet, "SURFconext," http://www.surfnet.nl/en/Thema/coin/Pages/ Default.aspx, visited Apr. 1, 2013.

[17] Internet2, "COmanage: Collaborative Organization Management," http://www.internet2.edu/comanage/, visited Apr. 1, 2013.

[18] GakuNin mAP: https://map.gakunin.nii.ac.jp/map/

[19] Takeshi Nishimura, Motonori Nakamura, Hitoshi Inoue, Kazutsuna Yamaji, Noboru Sonehara, "Group Management System in Access Federation for E-book Services", IPSJ SIG Technical Report, Vol.2011-IFAT-102, No.5, 2011/3.

[20] N. Sakimura, J. Bradley, M. Jones, B. de Modeiros, C. Mortimore, E. Jay, "OpenID Connect Messages 1.0 – draft 17", http://openid.net/specs/openid-connect-messages-1_0.html, 2013.