A Hybrid Model of Attribute Aggregation in Federated Identity Management

Md. Sadek Ferdous, Farida Chowdhury and Ron Poet

Accepted for publication in Chang V., Ramachandran M., Walters R., Wills G. (eds)
Enterprise Security (2017). Lecture Notes in Computer Science, vol 10131. Springer, Cham
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-54380-2_6

Abstract The existing model of Federated Identity Management (FIM) allows a user to provide attributes only from a single Identity Provider (IdP) per service session. However, this does not cater to the fact that the user attributes are scattered and stored across multiple IdPs. An attribute aggregation mechanism would allow a user to aggregate attributes from multiple providers and pass them to a Service Provider (SP) in a single service session which would enable the SP to offer innovative service scenarios. Unfortunately, there exist only a handful of mechanisms for aggregating attributes and most of them either require complex user interactions or are based on unrealistic assumptions. In this paper, we present a novel approach called the ***Hybrid Model*** for aggregating attributes from multiple IdPs using one of the most popular FIM technologies: Security Assertion Markup Language (SAML). We present a thorough analysis of different requirements imposed by our proposed approach and discuss how we have developed a proof of concept using our model and what design choices we have made to meet the majority of these requirements. We also illustrate two use-cases to elaborate the applicability of our approach and analyse the advantages it offers and the limitations it currently has.

1 Introduction

In the last fifteen years or so, we have seen a tremendous expansion of the Internet and web-enabled online services. To allow users to access different online services in a seamless manner while maintaining security and privacy, the concept of Identity Management (IdM, in short) has been introduced which resulted in various different Identity Management Systems (IMS, in short). Shibboleth (*Shibboleth*, 2016), OpenID (*OpenID Authentication 2.0 - Final*, 2007), Microsoft's CardSpace (Chappell, 2006), etc. are all examples of different IMS. Among different IMS, Federated Identity Management (FIM, in short) has gained much popularity. The FIM model is based on the concept of Identity Federation (also known as Federated Identities or Federation of Identities). Security Assertion Markup Lan-

M. S. Ferdous
Electronic and Computer Science, University of Southampton, UK, E-mail: S.Ferdous@soton.ac.uk

F. Chowdhury
Computing Science and Mathematics, University of Stirling, UK E-mail: fch@cs.stir.ac.uk

R. Poet
School of Computing Science, University of Glasgow, UK E-mail: Ron.Poet@glasgow.ac.uk

guage (SAML) (Standard, 2005) and OpenID (*OpenID Authentication 2.0 - Final*, 2007) are two of the most popular technologies used in identity federation.

Even though the federated services have improved the usability and experience of online services, there exists a serious limitation: a user can use only one IdP in one session (D. Chadwick & Inman, 2009). This is an unrealistic restriction that assumes that one IdP would be able to provide any number of user attributes to the SPs where in reality users have different attributes stored in different providers. To illustrate the restriction, let us consider a real-life scenario in the setting of FIM. Imagine a user, named Alice, wants to buy an age restricted product from her favourite online shop. She might need to use her Governmental IdP (e.g. Passport or Driving authorities who issue her passport or driving licence) to prove her age, her bank details to pay for the product and her loyalty card information to collect loyalty points during this purchase. All this information are stored at different IdPs. Even if we assume that the online seller has a federated agreement with all these IdPs, the current setting of federated services does not allow Alice to provide all this information in a single session (e.g. during her purchase). Allowing users to aggregate attributes from different providers will enable them to release these attributes to an SP in a single service session which could be used to provide innovative services.

The concept of attribute aggregation has been introduced to tackle this very problem. A number of novel approaches exist for aggregating attributes targeted for federated services. Even though each model has their own strengths, most of them suffer from serious limitations: many of them are based on either complex user interactions, unrealistic assumptions or have not been implemented in practice. In this paper, we present a hybrid approach for federated services which is based on two existing models and has been designed to specifically address these limitations.

The contributions of the paper are:

- We present our hybrid model based on SAML that does not require complex user interactions.
- We present a threat model for identifying different threats for our proposed system.
- We formulate a list of functional, security, privacy and trust requirements for our model. The security and privacy requirements have been devised as mitigation strategies against the identified threats.
- We provide an in-depth analysis of the design choices that we have made to ensure that most of these requirements are met.
- The current SAML specification does not have any mechanism to allow an SP to pass any information regarding the requested attributes from the IdPs during the authentication process. We have proposed a mechanism to rectify this problem.
- We discuss the developed proof of concept based on our proposals to illustrate the applicability of our model using two use-cases.
- Finally, we discuss the advantages that our model offers and the limitations it has.

The rest of the paper is structured as follows. In Section 2, we present a brief background of Identity, Federated Identity Management, Attribute Aggregation and SAML along with a few relevant mathematical notations and definitions that will be used throughout the paper. We provide a comparative analysis of existing attribute aggregation models in Section 3. We present our Hybrid Model along with a threat model for the proposed approach, highlight different requirements imposed by our model and discuss the design choices we have made to develop a proof of concept in Section 4. A few use-cases using our model are illustrated in Section 5. Section 6 analyses how our proof of concept meets the stated requirements, discusses the advantages that our model offers, the limitation it has and the scope of future work. Finally, we conclude in Section 7.

2 Background

In this section, we provide a brief background of Identity, Identity Management and Federated Identity Management. Then, we Mathematically define the term *Attribute Aggregation*. Finally, we briefly describe Security Assertion Markup Language (SAML).

2.1 Entity, Identity, Identity Management and Federated Identity Management

An entity is a physical or logical object which has a separate distinctive existence either in a physical or logical sense (Ferdous et al., 2009). Every entity has its own identity which is used to uniquely identify it in a certain context. In this paper, we are mostly interested about entities which digitally present users in an application system of an enterprise (organisation).

To define the identity of a user we utilise the Digital Identity Model (DIM) introduced in (Ferdous et al., 2014). According to this model, the (whole) identity of a user is actually distributed in different partial identities which are valid within a security domain (context) of an enterprise (organisation). Since the partial identity of a user is only valid within a domain, it is essential to specify the domain whenever a partial identity is mentioned. Each such partial identity consists of a number of attributes and their corresponding values, valid within the domain of a particular organisation.

Let us assume that D denotes the set of domains and d defines the domain of a single organisation whereas U_d denotes the set of users, A_d denotes the set of attributes and AV_d denotes the set of values for those attributes within d. Then, we can relate users and their attributes in a domain by the following partial function:

Definition 1 Let $\mathit{atEntToVal}_d : A_d \times U_d \to AV_d$ be the (partial) function that for an entity and attribute returns the corresponding value of the attribute in domain d.

The function is partial as not all entities have a value for each attribute. This also makes sense in practical systems as in many such systems, users are required to provide values for a number of attributes (e.g. email, telephone number, etc.). However, there remain some optional attributes (e.g. age, postal addresses, etc.) for which users may not provide any values.

Then, we can define the partial identity of a user (u) using the following definition.

Definition 2 For a domain d, the *partial identity* of a user $u \in U_d$ within d, denoted $\mathit{parIdent}_d^u$, is given by the set:

$$\{(a,v) \,|a \in A_d,\ \mathit{atEntToVal}_d(a,u) \text{ is defined and equals } v\}.$$

If we consider that there are n valid attribute-value pairs for a user u, the partial identity of u in d can also be defined as:

$$\mathit{parIdent}_d^u = \{(a_1,v_1),(a_2,v_2),(a_3,v_3)...(a_n,v_n)\}.$$

The (total/whole) identity of an entity can be defined as the union of all her partial identities in all domains.

Definition 3 For an entity $u \in U$, the *identity* of u is given by the set:

$$\mathit{ident}^u = \bigcup\{(d,\mathit{parIdent}_d^u) \,|\, d \in \mathit{DOMAIN} \text{ such that } u \in U_d\}.$$

The concept of Identity Management (IdM) has been proposed to facilitate the management of digital identities (Ferdous et al., 2009). Formally, IdM consists of technologies and policies for representing and recognising entities with their digital identities (Jøsang et al., 2007). A system that is used for identity management is called an Identity Management System (IMS). Each IMS involves the following parties:

Client/User. A client/user receives services from a service provider (see below). Any entity can be a client, however we assume that each client is a user.

Service Provider. A Service Provider (SP, in short) is an organisation that provides services to the clients or to other SPs. It is also known as the Relying Party. We will use the notation *SP* to denote the set of service providers.

Identity Provider. An Identity Provider (IdP, in short) is an organisation that provides digital identities to allow clients to receive services from a SP. We will use the notation *IDP* for the set of identity providers.

There are different models of identity management. One of the most widely used IdM models is the Federated Identity Management (FIM) which is based on the concept of Identity Federation. A federation with respect to Identity Management is a business model in which a group of two or more trusted organisations legally bind themselves with a business and technical contract (D. W. Chadwick, 2009, Ferdous et al., 2012). It allows a user to access restricted resources seamlessly and securely from other partners residing in different (identity/security) domains. The IdPs and SPs who bind themselves in such a way form the so-called Circle of Trust (CoT) which makes them a part of the same federation. One major advantage of the FIM is its Single Sign On (SSO) capability that allows users to log in to one IdP and then access services from autonomous and federated service providers without further logins, thus alleviating the need to log in every time a user needs to access those related systems. A good example is the Google Single Sign On service which allows users to log in to a Google service, e.g., Gmail, and then allows them to access other Google services such as Calendar, Documents, YouTube, Blogs and so on.

One of the functionalities of an IMS is to enable users to share their partial identities between different organisations (e.g. an IdP and SP). During this process, for the sake of privacy, users usually do not share their full partial identities between two organisations. Instead, a privacy-friendly approach is to share a limited view of a user's data across organisational boundaries whenever needed. Such a limited view is defined as the profile of a user. Mathematically, a profile is a subset of the partial identity of a user within a domain: $PROFILE_d^u \subseteq parIdent_d^u$. Hence, we can define the profile of a user $u \in U_d$ in domain d in the following way, where $j \leq n$:

$$PROFILE_d^u = \{(a_1, v_1), (a_2, v_2), (a_3, v_3)...(a_j, v_j)\}.$$

One of these attributes generally is an identifier which is used to identify a user in an IdP and as well as in an SP. To protect the privacy of a user, an IdP generally utilises two different types of identifiers. The first type is called a persistent identifier which is used for authenticating a user using a special attribute called a credential. In different systems, a persistent identifier takes different forms such as a username, email and phone number. On the other hand, passwords are the most widely used credentials nowadays. The second type of identifier is called a transient or pseudonymous identifier which an IdP generates for a user to be used in a specific SP. Such a pseudonymous identifier is embedded within the profile of the user so that the SP can maintain a session for the user. The main benefit of this approach is that one unique pseudonymous identifier for a user is valid to be used within a single SP, hence, two malicious SPs cannot collude together to track their users over two security

Table 1: SAML notations

SAML Authentication Request:	$AuthnReq = (id_{req}, id_{sp})$
SAML Assertion:	$SAMLAssrtn = (PROFILE^{u}_{idp})$
Encrypted SAML Assertion:	$EncSAMLAssrtn = (\{SAMLAssrtn\}_{K_{sp}})$
SAML Response:	$SAMLResp = (id_{req}, id_{sp}, id_{idp}, (\{SAMLAssrtn\}_{K^{-1}_{idp}} \mid \{EncSAMLAssrtn\}_{K^{-1}_{idp}}))$

domains. We will use the notation id_u to denote a persistent identifier of a user and id'_u to denote a pseudonymous identifier of the user. In addition, the notation $pass_{id_u}$ will be used to denote the credential for id_u.

2.2 Attribute Aggregation

Attribute Aggregation is the mechanism of aggregating or combining attributes of a user retrieved from multiple identity providers in a single session. As mentioned earlier, the profile of a user in an IdP contains the subset of attributes of that user retrieved from that IdP. Therefore, this means that attribute aggregation, in reality, is actually an aggregation of profiles of a user from different IdPs and the terms "profile" and "attribute" will be used interchangeably. Formally, the set of aggregated attributes can be defined in the following way:

For a user u, the set of aggregated attributes of u, denoted by $Att{-}Agg^{u}$ is given by:

$$Att{-}Agg^{u} = \bigcup \{PROFILE^{u}_{d} \mid d \in DOMAIN\}\,.$$

2.3 Security Assertion Markup Language (SAML)

SAML is one of the most widely deployed FIM technologies (Standard, 2005). It is an XML (EXtensible Markup Language)-based standard for exchanging authentication and authorisation information between different autonomous domains. It relies on a request/response protocol in which one party (an SP) requests particular identity information about a user and the other party (an IdP) responds with the information using an assertion. An assertion may contain three different types of statements: i) Authentication statements - which are used to state that a user has been authenticated by an asserted IdP at a specified time, ii) Attribute statements - which are used to specify the attributes of a user and iii) Authorisation statements - which are used to assert that a user is permitted a certain action on specified resources under certain conditions.

A SAML protocol flow between a user, an IdP and an SP is as follows. When a user tries to access a resource/service provided by the SP, if the user is already not authenticated and authorised, the SP forwards the user to a service called the *Discovery Service*, also known as the Where Are You From (WAYF) Service, where a pre-configured list of trusted IdPs is shown to the user. The user chooses her preferred IdP and then the user is forwarded to the IdP with a SAML authentication request. The authentication request consists of an identifier and the entity ID of the SP. We denote a SAML authentication request with *AuthnReq* and model it as presented in Table 1, where id_{req} denotes the identifier in each SAML request and id_{sp} denotes the identifier of the SP which is represented as the *entityID* in a SAML metadata (see below).

The IdP authenticates the user and a SAML response containing a SAML assertion is sent back to the SP. The assertion consists of the profile (user attributes) of the user as released by the IdP. We denote the assertion with *SAMLAssrtn* and model it as presented in Table 1. SAML also supports the notion of encrypted assertion where the assertion is encrypted with the public key of the SP. We denote an encrypted assertion with *EncSAMLAssrtn* and we model it as presented in Table 1, where K_{sp} represents the public key of the SP. The (encrypted or unencrypted) SAML assertion is at first digitally signed and then embedded inside a SAML response which also consists of the request identifier, the entity ID of the IdP and the entity ID of the SP. We denote a SAML response with *SAMLResp* and model it as presented in Table 1, where id_{req} denotes the SAML request identifier whereas id_{idp} and id_{sp} denote the entity ID of IdP and the SP respectively. Furthermore, $\{SAMLAssrtn\}_{K_{idp}^{-1}}$ represents the a digitally signed assertion with the private key of IdP (K_{idp}^{-1}). On the other hand, $\{EncSAMLAssrtn\}_{K_{idp}^{-1}}$ represents a digitally signed encrypted assertion encrypted with the public key of the SP (K_{sp}) and signed with the private key of the IdP (K_{idp}^{-1}).

When the SP receives the response, it extracts the (encrypted/unencrypted) SAML assertion. If it contains an encrypted assertion, the assertion is decrypted with the private key of the SP (K_{sp}^{-1}) and then its signature is validated with the public key of the IdP (K_{idp}). If the response contains an unencrypted assertion, its signature is validated using the public key of the IdP. If the signature is valid, the SP retrieves the embedded user attributes (the profile, $PROFILE_{idp}^{u}$) from the assertion and takes an authorisation decision.

To enable the protocol flow discussed above, a notion of trust needs to be established between an IdP and an SP inside a federation. This notion of trust between the IdP and SP in SAML is established by exchanging the respective metadata of the IdP and SP and then storing such metadata at the appropriate repositories. This enables each party to build up the so-called Trust Anchor List (TAL). This exchange takes place in an offline fashion after a technical contract between the IdP and SP is signed. Moreover, this has to be done before any interaction takes place between the said IdP and SP. Once the exchange of metadata is complete, the IdP and SP are said to be part of the same federation (the CoT). A metadata is an XML file in a specified format that contains:

- an entity descriptor, known as *entityID* which represents an identifier for each party,
- service endpoints (the locations of the appropriate endpoints for an IdP and an SP where the request will be sent to and where the response will be consumed respectively),
- a certificate, containing the public key of the party,
- an expiration time, and
- contact information.

Metadata serves three purposes as explained below.

- Firstly, it allows each entity to discover the required service endpoint of another entity for sending SAML requests/responses.
- Secondly, the embedded certificate can be used by an SP to verify the authenticity of any SAML Assertion.
- Thirdly, metadata behaves like an anchor of trust for each party. During the discovery service at an SP, the list of IdPs returned to the user only contains those IdPs whose metadata can be found in its repositories (in other words in the TAL) and only those IdPs are considered trusted. An SP will allow a user to select only those IdPs which are trusted. Similarly, an IdP will respond only to those requests that are initiated from an SP whose metadata can be found in its TAL.

SAML is a widely-used FIM technology in higher education and Governmental online services. It has many implementations such as Shibboleth (*Shibboleth*, 2016), SimpleSAMLphp (SSPHP, in short) (*SimpleSAMLphp*, 2016) and ZXID (*ZXID*, 2016), however, only Shibboleth and SimpleSAMLphp are widely used. At first, we are interested to see if any particular implementation of SAML allows a specific set of attributes to be requested (denoted as *Attribute Request*), if the implementation allows the user to select and release specific attributes (denoted as *Attribute Release*), known as the Selective Disclosure in the Identity

Table 2: Comparison of protocols and implementations

Protocol	Implementation	Attribute Request	Attribute Release	Attribute Aggregation
SAML	Shibboleth	✓	✗	✗
	SSPHP	✓	✓	✗

Management terminologies, and if the implementation allows any mechanism of attribute aggregation (denoted as *Attribute Aggregation*).

How and what attributes are requested and released in SAML depend on a specific implementation. Also, how attributes are selected is not standardised in SAML and depends entirely on the specific implementation. For example, a Shibboleth SP requests attributes from an IdP using the SAML *AttributeQuery* request and the attributes are released based on an attribute release policy (ARP) (Cantor, 7 January, 2008). On the other hand, SimpleSAMLphp also supports the *AttributeQuery* feature, however, this request cannot be used during the authentication process. Nonetheless, SimpleSAMLphp has a built-in module called *Consent* which allows selective disclosure of attributes.

Even though there are available frameworks that allow integrating different social networks with SAML federations (D. W. Chadwick et al., 2011) to offer federated services, the SAML specification has no mechanism to aggregate attributes nor does any SAML implementation provide this facility. However, the implementations of existing models of attribute aggregation are mainly based on SAML. The comparison between Shibboleth and SimpleSAMLphp for Attribute Request, Attribute Release and Attribute Aggregation is presented in Table 2. In the table, the symbol ✓ has been used to indicate that the particular implementation supports that particular function and the symbol ✗ indicates the absence of support.

3 Existing Attribute Aggregation Models

There are several existing attribute aggregation models. In the following sections, we provide a brief description of each model.

Application Database (AD) Model. This is the simplest form of attribute aggregation model (Klingenstein, 2007, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011). In this model, an SP might store additional user attributes such as a local identifier, user-preferences for that particular service, group membership, etc., in addition to the attributes supplied by an IdP. The SP creates a mapping of the SP-created identifier to the IdP-supplied identifier to store these additional attributes, into a local repository. Such local attributes can be retrieved later using this mapping to determine if the user is authorised to access a particular service.

SP-Mediated (SPM) Model. In this model, an SP allows a user to aggregate attributes from multiple IdPs in a single session (Klingenstein, 2007, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011). The user is forwarded to different IdPs one after another where she is authenticated separately. Then she is returned back to the SP with the IdP-supplied attributes. The SP combines the sets of attributes at its end to determine if the user can access a particular service.

Linking Service (LS) Model. Linking Service model is a combination of the linking and identity relay model (see below). It consists of a special type of SP called the Linking Service (LS) (D. Chadwick & Inman, 2009, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011) which is used by a user using a LS-supplied identifier. This identifier is used to link different IdPs using the IdP-supplied LS-specific persistent identifiers in a table called the Linking Table. To access any service of an SP, the user visits the SP and is forwarded to the first IdP. The user authenticates at the first IdP and then an assertion containing user-attributes, the persistent identifier for the LS and a reference to the LS is returned to the SP. The SP forwards the persistent identifier to the LS to aggregate attributes. At this point, two options are available: either the LS can retrieve the list of linked IdPs for this persistent identifier using the Linking Table and retrieve attributes from each of them which are then combined at the LS and is returned to the SP. Alternatively, the LS can send back the list of linked IdPs to the SP. The SP, then, retrieves attributes from each IdP. Based on the aggregated attributes, the SP determines if the user can access the service.

Identity Federation/Linking (IFL) Model. This model, introduced by the Liberty Alliance framework, is one of the first models to address the problem of attribute aggregation (Klingenstein, 2007, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011). In this model, IdPs allow a user to create a pair-wise link between two IdPs. To create the link, the user has to visit and authenticate to the first IdP. The first IdP will ask the user if she wants to federate this IdP with another IdP. If chosen, the user will be asked to federate the second IdP with the first one. At this point, both IdPs will interact with each other to create a random alias. During accessing services from an SP, one IdP will provide that random alias to the SP along with the assertion containing attributes. The SP can use that alias to retrieve another assertion containing attributes from the other IdP. Combining attributes from both IdPs, the SP can determine if the user can access a service.

Identity Proxying (IP) Model. In this model, an SP allows a user to aggregate attributes from multiple IdPs using a trusted IdP known as the *Proxy IdP* (Klingenstein, 2007, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011). The user is forwarded to the Proxy IdP at first and then the trusted IdP forwards the user to other multiple IdPs. After the user is authenticated separately at each IdP, the user returns back to the trusted IdP with an assertion including attributes. At this point, the trusted IdP validates each assertion, retrieves attributes from them and combines all these attributes. The trusted IdP might supplement the combined set with its own user-attributes and then reasserts all attributes to the SP as its own attributes. The SP, not being aware of other IdPs, assumes that all attributes have been released by the trusted IdP. Based on the combined attributes, the SP determines if the user can access the service.

Identity Relay (IR) Model. The Identity Relay model is a generalised case of the Proxying model (Klingenstein, 2007, Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011). Since the Proxying model requires an SP to have a strong trust in the trusted IdP, it cannot function properly in situations when the Proxy IdP cannot be fully trusted. The Identity Relay model fits in such scenarios where an intermediary IdP, known as the *Relay IdP*, is used instead of a Proxy IdP. A user is forwarded to the Relay IdP at first and then the Relay IdP forwards the user to other multiple IdPs. The user is authenticated separately at each IdP and is returned back to the Relay IdP with encrypted assertions containing user-attributes. The encrypted assertions are encrypted with the public of the SP so that only the SP can decrypt them. The Relay IdP combines all encrypted assertions into a single assertion and forwards it to the SP. The SP extracts embedded encrypted assertions from this assertion, decrypts and validates each assertion one by one to retrieve attributes from other IdPs. Based on the combined attributes, the SP determines if the user can access the service.

Client-Mediated (CM) Model. This model is similar to the Relay Model. Here, the functionality of the Relay IdP has been replaced by an intelligent user-agent or application that has the capability to aggregate attributes from different IdPs (Klingenstein, 2007, Bob

Table 3: Aggregation Models: Strengths and Weaknesses

Models	Advantages	Disadvantages
AD	Easy to implement and maintain. Small number of requirements.	Aggregation from only one IdP in a session. Changing identifiers causes the system to fail.
SPM	Aggregation from a number of IdPs in a session.	Hard to maintain. Multiple logins at the IdPs in a single SP session. No implementation exists.
LS	Secure and Privacy-preserving. Proof of concept implementation available. Attribute aggregation from multiple IdPs in a single session. SSO capability during service access.	Unrealistic trust assumption. Hard to deploy and difficult to maintain. The LS represents a single point of failure.
IFL	Secure and Privacy-preserving. Proof of concept implementation available. SSO capability during service access.	Unrealistic trust assumption. Difficult to maintain and scale. Attribute aggregation from only two IdPs in a single session.
IP	Attribute aggregation from multiple IdPs in a single session. The SP is easy to maintain. Relatively small number of requirements.	The Proxy IdP requiring huge trust might not be suitable. The Proxy IdP is the single point of failure. No implementation exists.
IR	Attribute aggregation from multiple IdPs in a single session. The SP is easy to maintain. Relatively small number of requirements. Less trust on the Relay IdP.	The Relay IdP is the single point of failure. Hard to maintain. No implementation exists.
CM	No need to rely on external IdPs. Attribute aggregation from multiple IdPs in a single session. Relatively small number of requirements.	No implementation exists currently. Extensive changes are required to ensure that the IdPs, the SPs and the client can interact.

Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, 2011) and hence the protocol flow of this model is similar to the Relay model. An SP informs the client about the IdPs that it trusts. The client forwards the user to each of these IdPs. After respective authentication at each IdP, the client receives assertions from all IdPs and present the combined set of assertions to the SP. The SP validates each assertion, retrieves all attributes and then determines if the user can access the service.

3.1 Analysis

Each of these models has their own strengths and weaknesses which is hard to comprehend at the first instance from the textual description as outlined above. Therefore, we present our analysis in tabular formats in Table 3 which outlines a side-by-side comparison of these models (Ferdous & Poet, 2013a). The more requirements one model has to fulfil the more complex it will be to establish, maintain and scale. Therefore, an optimal aggregation model should have a relatively small number of requirements with a good number of advantages. It is clear from Table 3 that no model is unquestionably superior to other models. Next, we choose a suitable candidate for further research.

At the beginning, our main goal has been to investigate if there is any way we can leverage the advantages of different models by combining them together to come up with a hybrid model that offers less complex user interactions, has a significantly smaller number of requirements as well as relatively fewer disadvantages and preferably has already been implemented. There is no direct available implementation of the AD, however, the model can be implemented with a little bit of additional code that will create, store and map the

IdP-supplied identifier with the local identifier using any existing SAML implementation. Unfortunately, its main limitation is that it can aggregate attributes only from one IdP and therefore it has been ruled out of our consideration. Similarly, the CM model does not have any implementation and it will require a considerable number of changes in the existing standards to ensure that different IdPs, SPs and users can interact using this model. The next two models that have existing implementations are the LS and the IFL models. Even though, both these models, especially the LS model, have a good number of advantages, their weaknesses (in Table 3) limit their usefulness and hence we have also ruled them out of our consideration. The SPM model allows an SP to aggregate attributes from multiple IdPs, however, to deploy this model, a significant number of changes (e.g. saving and restoring sessions while the SP makes the SAML authentication request and receives the response, how the WAYF page displays the list of IdPs, etc.) are required at the SP side. We do not want an SP needing to make a vast number of changes to ensure that the existing SAML SPs are as less affected as possible. With all these goals in mind, the IP and the IR models seem to be the suitable candidates for our hybrid model. They both offer a good number of advantages and require a small number of trust assumptions. Moreover, one significant disadvantage of the IP model, the huge reliance on the Proxy IdP, can be tackled by allowing the Proxy IdP to act as the Relay IdP when it is required which will reduce the amount of trust needed on that IdP. This essentially brings us to this question: can we combine these two models to leverage the advantages in such a way that the advantages of one model can be used to reduce the disadvantages of the other model and then offer something more that none of the other existing models offer individually? In this work, we will seek answer to this very question.

4 The Hybrid Model

With the IP and the IR models chosen as the suitable candidates for our research, the next thing we would like to do is to design a model by combining the IP and IR model which will be denoted as the ***Hybrid Model*** and the trusted IdP of our model will be denoted as the ***Hybrid IdP***. At first, we devise a threat model for identifying threats in our proposed model and for formulating different security and privacy requirements to minimise these threats.

4.1 Threat Modelling

Threat modelling is an integrated process of designing and developing a secure system. A well-defined threat model helps to identify threats on different assets of a secure system. In order to tackle such threats, different mitigation strategies need to be outlined by formulating security and privacy requirements (Myagmar et al., 2005). In essence, a threat modelling consists of the following steps (Desmet et al., 2005, De Cock et al., 2005):

1. listing assets of the system,
2. identifying possible threats on those assets and
3. outlining mitigation strategies.

Each secure system since has different assets, the threat modelling process of one system will be considerably different than that of another system. Here, we would like to restrict our focus on modelling threats in the setting of a FIM System. There exist only a very few papers in this context; notably in (Khattak et al., 2010, Dominicini et al., 2010). Additionally, there are a few papers focusing on the issue of threat modelling in the setting of web services (Desmet et al., 2005, De Cock et al., 2005). Based on these works, each single step of our threat modelling process is described in the following sections.

4.1.1 Listing Assets

An asset is the abstract or physical resource in a FIM system that needs to be protected from an adversary (attacker) (Myagmar et al., 2005). It is the resource for which a threat exists and represents the target of the adversary in the system. Examples of physical assets include different hardware such as the server machine where a specific system is hosted, the network components such as routers by which data is transmitted from one system to another and machines such as computers, smartphones and tablets from which users access different online services. Examples of abstract resources include different software such as operating systems running the physical servers, web servers, web services and data. Since our focus in this paper is on federated web services, we are mostly interested about software resources. The motivation behind this step is to highlight those assets in the system which can be the target of an adversary so that associated threats for these assets can be identified. Based on the threat modelling of an Identity Management System in (Dominicini et al., 2010), we list the following assets:

- **Partial Identity of a user**. In a FIM system, the partial identity ($parIdent_d^u$) of a user u in a domain/system d can be regarded as the core asset since the main purpose of such a system is to help a user to manage her partial identity and leverage it for accessing online services.
- **Activities associated with a partial identity**. Not only the partial identity, but also how the partial identity of a user is leveraged for accessing different online services using a FIM system represents a valuable asset. This is because such associated activities offer a lucrative way for an adversary to track users across different application domains.
- **Web services**. In a FIM system, web services must be protected from unauthorised accesses and should be accessible to only those who can prove their rights to access them using their partial identities.

4.1.2 Identifying Threats

A threat represents the activity or capability of an adversary onto an asset of a system with an intention to invade the security of the system or invade the privacy of a user in the system (Myagmar et al., 2005) . The main motivation behind this step is to identity possible threats on different assets of the system so that proper mitigation techniques can be carefully planned. Based on the treat modelling process presented in (Desmet et al., 2005, Dominicini et al., 2010, Khattak et al., 2010), we identify the following threats:

1. **Spoofing**. Information regarding the partial identity of a user is disclosed to another unauthenticated and unauthorised user allowing the second user to impersonate the first user.
2. **Tampering**. An attacker can intercept data while being transmitted and change it with malicious intent.
3. **Repudiation**. An attacker can deny performing a certain action and another party cannot prove it.
4. **Information disclosure**. User attributes are disclosed to service providers without the respective user's knowledge and consent.
5. **Escalation of privilege**. Aggregated attributes are released to an SP in such a way that it will enable an attacker to gain unprivileged accesses.
6. **Malicious providers**. Malicious service providers getting hold of unnecessary attributes can abuse them for their own benefit. Two or more malicious service providers can collude to track the activities of a user in order to build a profile of the user without her knowledge. A Hybrid IdP can abuse the aggregated attributes by storing them at its end.

7. **Lack of control**. Once attributes of a user is released to an SP, users have no control over them and the SP can abuse the attributes without any knowledge of the user.

4.1.3 Mitigation Strategies

Once the threats have been identified, mitigation strategies must be carefully planned and transformed into actions to minimise the threats as much as possible. One way to achieve this is to transform them into requirements which the system must fulfil as proposed in (Myagmar et al., 2005). We have adopted this approach. In the next subsection, we formulate several security and privacy requirements which act as the mitigation strategies and fulfilling them will ensure that threats are minimised.

4.2 A Study of Requirements

Next, we formulate a set of Functional, Security, Privacy and Trust requirements that we want our model to fulfil. The requirements are based on the requirements for the IP and IR models presented in (Ferdous & Poet, 2013a) and have been rephrased with the reference of the Hybrid IdP. Among them, the security and privacy requirements formulated here act as mitigation strategies to minimise threats identified above as mentioned earlier. The requirements are presented below.

Functional Requirements (FR): The functional requirements ensure that a system behaves as desired. The requirements are:

F1. The Hybrid IdP and the SPs are part of the same federation. The other SAML IdPs and the Hybrid IdP are part of the same federation. The other SAML IdPs are part of the same federation with an SP, needed only for the IR model.

F2. A session is maintained at the Hybrid IdP so that the Hybrid IdP can correlate the attributes from the current IdP with attributes retrieved previously from other IdPs.

F3. The Hybrid IdP has dual capabilities of an IdP as well of an SP. The Hybrid IdP has to act as an IdP to the SP and as an SP to other IdPs.

F4. For the IP model, the assertion returned by other IdPs should be targeted for the trusted (Hybrid) IdP so that it can validate each assertion, extracts attributes from them and then aggregates all of them, possibly also with its own attributes.

F5. For the IR model, the assertions returned by other SAML IdPs should be targeted for an SP. The Hybrid IdP will just aggregate all assertions and embed them inside another assertion and send it back to the SP. The SP will validate the outer assertion and retrieve all embedded assertions. Then it must validate each assertion in turn to extract attributes from them.

Security Requirement (SR): The security requirements ensure the security during an interaction between entities. These requirements have been formulated to undermine the threats outlined above. The requirements are:

S1. The IdP should have a secure mechanism for user registration and authentication. A user must be authenticated before she can access her attributes and release those attributes to an SP. This requirement undermines threat 1.

S2. The transmitted data between two parties is not disclosed to any unauthorised entity.

S3. The transmitted data is not altered during transmission. Combinedly, S2 and S3 undermine threat 2.

S4. A user, once committed for a transaction, cannot deny her commitment in the transaction. It undermines threat 3.

Privacy Requirements (PR): The following privacy requirements ensure that the privacy of a user is preserved during attribute aggregation via any FIM system.

P1. The Hybrid IdP should allow a user to access services anonymously or using pseudonymous identifiers. This will ensure that two malicious SPs cannot collude to build a profile of a user.

P2. The Hybrid IdP should implement the selective disclosure of attributes to allow a user to select specific attributes before they are released to an SP. By this way, the user can choose specific attributes for a specific SP.

P3. The Hybrid IdP should allow a user to provide explicit consent before any data is released to an SP. Combinedly, P2 and P3 undermine threat 4.

P4. To ensure data minimisation, an SP must inform a user about the minimum number of attributes that the user must release to access any particular service of that SP.

P5. One way to enforce the control over the data, released to an SP, is to allow users to administer their data remotely, preferably using remote policies. If an IdP and an SP inside a federation offer this facility, there should be a user-interface at the IdP for the user to administer such policies and there should be a mechanism to exchange such policies between the IdP and the SP. This undermines threat 7.

Trust Requirements (TR): This set enlists those requirements that outline the scopes in which each entity trusts another entity during an attribute aggregation scenario using a FIM system.

T1. A user and an SP trust that an IdP has implemented satisfactory user registration procedures and authentication mechanisms. The SP trusts that the IdP will authenticate the user appropriately as per the requirement and will release user attributes securely.

T2. A user trusts that an IdP protects the user's privacy to an SP by using anonymous or pseudonymous identifiers and the IdP will release only those attributes to the SP that the user has consented to.

T3. A user trusts that an SP will ask only for the minimum number of user attributes that are required to access any of its services and will not abuse the released user attributes.

T4. The IdP and the user trust that the SP adheres to the agreed privacy policies regarding non-disclosure of user data.

T5. For the IR model, the user trusts that the other SAML IdPs release assertions in such a way that they are only accessible by the corresponding SP.

In essence, these functional, security and privacy requirements signify the technical conditions that must be considered while designing and developing an attribute aggregation mechanism using a FIM system and can be used to satisfy the trust requirements. Now, to design a model to aggregate attributes from other SAML IdPs, it will require formulating novel requirements. We have identified the gaps in the IP and IR models and formulated them as requirements which are enlisted below:

Functional Requirements (FR):

F6. There should be a mechanism to filter out an IdP once a user has aggregated attributes from that IdP to ensure that the user does not chose the same IdP over and over again.

F7. The existing proxy model assumes that the aggregated attributes, from the SAML IdPs, are combined to build a single assertion to pass over to an SP. The effect of this assumption is that the original sources of the attributes are lost, leaving the SP with the notion that the attributes have originated from the trusted Proxy IdP. This is a risky assumption to make which might escalate privileges since some (or even all) attributes might originate from the SAML IdPs which the SP might not trust at all. To prevent the SP from making such risky assumptions, the Proxy IdP must indicate the source of all attributes. Moreover, the Proxy IdP should use the NIST LoA (Level of Assurance or Level of Authentication (NISTWP, 2006)) of level 1 to 4, where the level 1 signifies the lowest and the lever 4 signifies the highest level of assurance, to assert the assurance level of the attributes from each IdP. The best way to embed these values is to group attributes from a single IdP and then insert the source and the perceived LoA value for each group.

F8. An SP should have the capabilities to interpret such groupings to ensure that the SP can easily identify the source of each attributes as well as the LoA value associated with them.

F9. It is the user who should have the ability to choose between the IP and IR models. If the user chooses the IR model, it means that she wants the other IdP to release an encrypted assertion, instead of the regular unencrypted assertion, to an SP via the Hybrid IdP. In such cases, the Hybrid IdP should forward the information regarding the SP to the other IdP to enable it to generate the encrypted assertion properly.

Security Requirements (SR):

S5. To ensure the security of attributes, the Hybrid IdP must not store the aggregated attributes (assertions for the IR model) once the session with the Hybrid IdP is terminated. Combinedly, P1, P4 and S5 undermine threat 6.

S6. The LoA of other IdPs are properly determined by the Proxy IdP. This helps an SP to take correct authorisation decision and hence undermines threat 5.

Privacy Requirements (PR):

There are no additional privacy requirements other than *P1 - P5*. However, it should be noted that all privacy requirements are applicable for external IdPs (including the Hybrid IdP).

Trust Requirements (TR):

All mentioned trust requirements for the IP and IR models are applicable for our model. Note that *T4* is applicable for the Hybrid IdP as well. In addition, we have the additional following requirements:

T6. A user trusts that the Hybrid IdP does not store any released attributes or assertions from the other IdPs.

T7. An SP trusts that the Hybrid IdP groups the attributes correctly according to their sources and the source for each group as well as the associated LoAs for each group are released to the SP.

4.2.1 Analysis

The trust requirement *T1* can be satisfied by fulfilling requirements *S1*, *S2* and *S3* since these security requirements ensure that the IdP has implemented the required methods for adequate user registration and subsequent authentication. The requirement *T2* can be satisfied by fulfilling *P1* and *P3* as these privacy requirements will enable a user to access services anonymously/pseudonymously and to provide explicit consent before releasing attributes to an SP. The requirement *T3* is fulfilled by *P4* as *P4* ensures data minimisation. Since fulfilling *P5* will enable a user and an IdP to determine how data are treated at an SP and if the SP adheres to the agreed policy, this will ensure *T4* being satisfied. Furthermore, if *F5* is satisfied, this will ensure that *T5* is fulfilled. Since meeting *S5* will ensure that aggregated attributes or assertions are not stored in the Hybrid IdP, this will fulfil *T6*. Finally, satisfying *F8* and *S6* will ensure that attributes are grouped according to their sources and the Hybrid IdP adds the correct LoA value for each source, this will fulfil the final trust requirement *T7*.

Table 4 illustrates a side-by-side comparison of requirements met by the IP, IR and Hybrid models. As mentioned earlier, the novel requirements of the Hybrid model are formulated to address the gaps in functionalities in the IP and IR models.

Table 4: Comparison of requirements

Models	**TR**	**FR**	**SR**	**PR**
IP	T1 - T4	F1 - F3, F4	S1 - 54	P1 - P5
IR	T1 - T5	F1 - F3, F5	S1 - S4	P1 - P5
Hybrid	T1 - T7	F1 - F9	S1 - S6	P1 - P5

4.3 Architecture

The architecture of our Hybrid Model is illustrated in Figure 1. The dotted area in the figure represents the Circle of Trust (i.e. the federation) between two entities. The Hybrid IdP, which acts as the trusted third party (TTP), takes the central stage in our model. Any number of SAML IdPs can be added with the Hybrid IdP. These IdPs will act as the third party IdPs and a user can aggregate attributes from these IdPs using the Hybrid IdP. Any such third-party SAML IdPs (*SAML IdP1* and *SAML IdP2* in Figure 1) must be federated with the Hybrid IdP. Furthermore, an SP has to be federated with the Hybrid IdP (as illustrated in Figure 1). Thus, the Hybrid IdP takes the role of a Proxy IdP and the protocol flow for any use-case will be similar to that of the IP model.

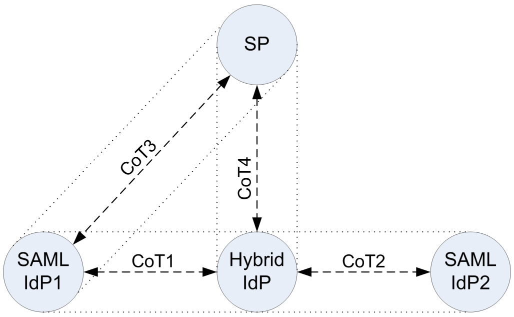


Fig. 1: Architecture of Hybrid Model.

One major limitation of this approach is that attributes are exposed to the Hybrid IdP during the aggregation process and hence, the Hybrid IdP needs to be trusted. Unfortunately, even if it is trusted, there is no guarantee that it will not abuse those attributes. One way to tackle this limitation is to enable the IR mode of the Hybrid IdP so that it acts as a Relay IdP. This will enable other SAML IdPs to release encrypted assertions to SPs via the Hybrid IdP. Since, all assertions will be exposed to the Hybrid IdP in encrypted formats, attribute will not be exposed. To enable this protocol flow, the SP must be federated with any other third-party SAML IdPs so that they can release encrypted assertions for the SP. The absence of any federation between an SP and any third-party SAML IdPs will prohibit the user to release any encrypted assertions from those IdPs to the SP via the Hybrid IdP. For example, in Figure 1, the SAML IdP1 can release encrypted assertions to the SP whereas the SAML IdP2 cannot.

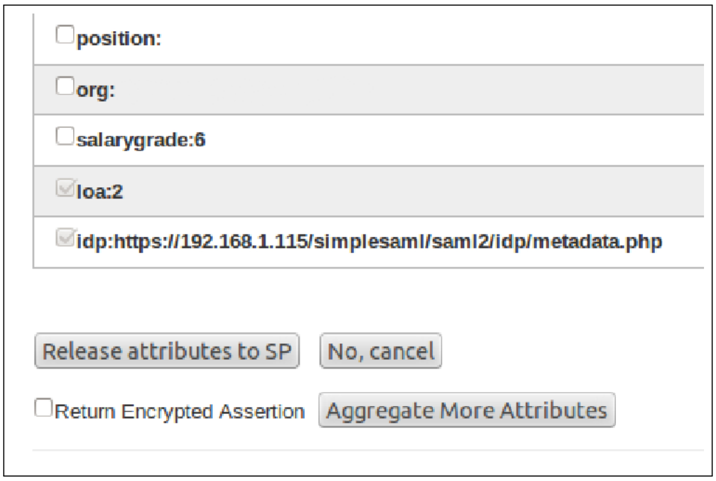
The whole architecture has been designed in such a way that it satisfies the majority of the requirements enlisted above. We will explain how our design choices achieve this in the subsequent section.Then, we will illustrate two use-cases for two different scenarios using our implementation.

4.4 Implementation

We have based our work using SimpleSAMLphp. This is for two reasons. Firstly, SimpleSAMLphp has built-in support for the Selective Disclosure of attributes based on its *Consent* module. Secondly, SimpleSAMLphp also has another built-in module called the *Multiauth* module that allows a SAML IdP to delegate the authentication task to an external SAML IdP and then release the attributes to an SP via a SAML assertion once the user has authenticated at the external IdP and returned back to the SAML IdP with attributes. This particular behaviour essentially allows the IdP to act as the Proxy IdP. However, it does not allow choosing more than one IdP for a particular session, hence attribute aggregation from multiple IdPs are not possible and cannot return an encrypted assertion to an SP, hence, it cannot perform as a Relay IdP. Even with these limitations, these two modules would give us a solid platform to base our work with the main focus to confine most of our changes to the Consent and the Multiauth modules and to modify the core code-base of SimpleSAMLphp only when there is an absolute necessity.

In SimpleSAMLphp, the consent module is loaded to show a consent form once a user is authenticated at an IdP. The consent form displays the list of attributes (the partial identity) of the user stored at the IdP and allows the user to choose the attributes that she wishes to release to an SP. We have modified the Consent module by adding two new inputs - a check box and a button - so that the consent form acts as the entry point for allowing users to aggregate attributes. The modified consent form is illustrated in Figure 2. The check box with the text *Return Encrypted Assertion* is used to toggle between the IP and IR mode and the *Aggregate More Attributes* button is used to initiate the process of attribute aggregation.

The original Multiauth module enables an IdP to act as a Proxy IdP to delegate the authentication to other IdPs. A list of such IdPs is shown to the user at the IdP. Once the user selects the IdP, she is forwarded to the chosen IdP where the user authentication takes place. Then the user is redirected back to the Proxy IdP where a session is created. Once the session is created, the user cannot choose another IdP without logging out from the Proxy IdP. We need to change this behaviour of the Multiauth module and the SimpleSAMLphp IdP code-base to allow users to choose other IdPs even if there is a session at the IdP. For this, we have made two amendments: i) the Multiauth module has been modified so that it shows the list of external IdPs, excluding those IdP(s) a user has already authenticated at, when the user clicks the *Aggregate More Attributes* button at the consent page and ii) the SimpleSAMLphp



□ position:

□ org:

□ salarygrade:6

☑ loa:2

☑ idp:https://192.168.1.115/simplesaml/saml2/idp/metadata.php

Fig. 2: The modified consent form.

IdP code-base has been modified to allow the user to initiate an authentication process even if the user has a session at an IdP and to retain that session when the user is redirected back from the chosen external IdP. This allows the IdP to collate attributes from the recently authenticated IdP with the attributes aggregated previously.

When attributes are aggregated from multiple sources at the Hybrid IdP and passed to an SP inside a single assertion, it will be impossible for the SP to identify the source of each single attribute. Without identifying the source of each attribute, it might be difficult for the SP to take access control decisions. None of the previous implementations considered this crucial issue. The best way to retian the source of the attributes is to group them according to their sources while aggregating them at the Hybrid IdP. In addition, the Hybrid IdP can also determine the LoA value for each IdP and add this information to each group. We need a data structure to hold all this information and one of the options for such a data structure is a list of lists where each entry of the list will signify a list holding the attributes from a particular IdP along with its LoA value. The structure is illustrated in Figure 3.

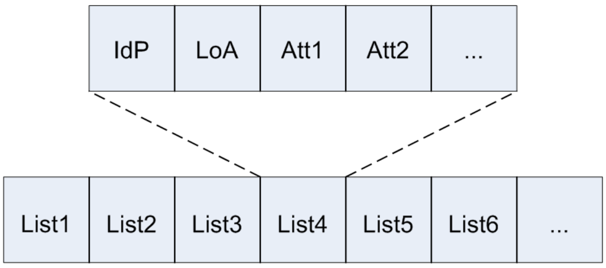


Fig. 3: The Data Structure for Attribute List.

In the current SAML specification, each attribute, with its value, is inserted inside an AttributeStatement element (Standard, 2005). The schema for the AttributeStatement is given in Listing 1. If the grouped attributes following our data structure are inserted in the AttributeStatement element in this manner, there are chances that each group might get mixed up with each other. To mitigate this problem, we propose the inclusion of a new SAML element called ***AttributeStatements***. The schema of the proposed element is given in Listing 2. The *AttributeStatements* element will essentially accommodate one or more AttributeStatement element(s) and will signify our proposed data structure. In this way, the Proxy IdP can include the IdP and LoA information for group of attribute inside each AttributeStatement element. We have modified the IdP and the SP code-bases of SimpleSAMLphp to reflect this mechanism.

The next amendment made is to handle the IR mode. The relay mode will enable the external IdP to release an encrypted assertion to the Hybrid IdP. To do so, we need two things: i) a mechanism to signal to an external IdP that the Hybrid IdP (actually the user) is requesting an encrypted assertion and ii) the Entity ID of an SP so that the external IdP can encrypt the assertion that is only decryptable by the SP. Since the SP is only communicating with the Hybrid IdP, not with the external IdP, the Hybrid IdP has the responsibility of passing on these two pieces of information. As stated earlier, a check box in the consent form is used to toggle between the IP and IR mode to indicate if a user wants to receive an encrypted assertion. Once the checkbox is ticked and the user clicks the *Aggregate More Attributes* button, an attribute called ***EmbedAssertion*** is added to the SAML authentication request which contains the Entity ID of the SP.

When the external IdP receives a SAML authentication request, it checks for an attribute called ***EmbedAssertion***. If this attribute is not found, the IdP behaves as usual. However, if the attribute is found, the external IdP knows that it has to release an encrypted assertion. Then, the Entity ID of the SP is retrieved from that attribute which is used to create an encrypted assertion. The encrypted assertion is then Base64 encoded and added as the value of a special type of attribute called ***encryptedAssertion*** into an AttributeStatement element which is then embedded inside a regular unencrypted SAML assertion. This regular assertion is then passed to the Proxy IdP. After receiving an assertion from the external IdP, the Hybrid IdP validates it and if it finds that there is an attribute called ***encryptedAssertion***, it is treated specially (see below). Otherwise, the attributes retrieved from the external IdP are aggregated with the previously aggregated attributes and are presented in the consent form. We have added necessary amendments to SimpleSAMLphp code-bases to reflect this behaviour.

Listing 1: Schema for AttributeStatement Element

```
<element name="AttributeStatement"
type="saml:AttributeStatementType"/>
<complexType name="AttributeStatementType">
  <complexContent>
  <extension base="saml:StatementAbstractType">
    <choice maxOccurs="unbounded">
      <element ref="saml:Attribute"/>
      <element ref="saml:EncryptedAttribute"/>
    </choice>
 </extension>
 </complexContent>
 </complexType>
```

Listing 2: Proposed Schema for AttributeStatements Element

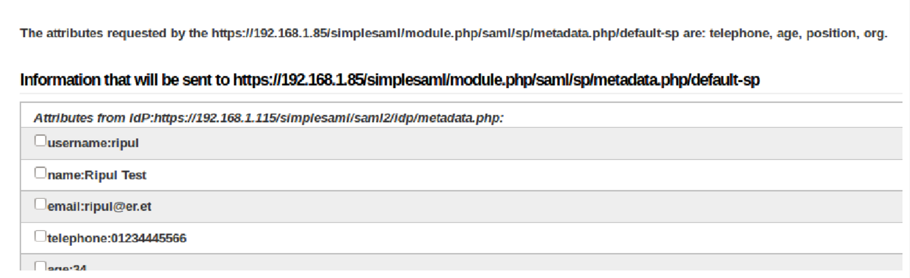
```
<element name="AttributeStatements"
type="saml:AttributeStatementsType"/>
```

```
<complexType name="AttributeStatementsType">
  <complexContent>
  <extension base="saml:StatementAbstractType">
    <choice minOccurs="1" maxOccurs="unbounded">
      <element ref="saml:AttributeStatement"/>
    </choice>
 </extension>
 </complexContent>
 </complexType>
```

To achieve data minimisation, the selective disclosure of attributes and to provide an optimal way of aggregating attributes, the user must know beforehand the attributes required by an SP to access its services. As mentioned earlier, one way to achieve this is to show the list of attributes for each specific service at the SP home page as adopted in (D. Chadwick & Inman, 2013). However, it is not realistic to assume that a user will remember such a list for each service. A draft to extend the SAML AuthnRequest element has been proposed in (Sampo Kellomäki, 2008) to include a SAML AttributeQuery statement within the authentication request. We have adopted this approach in this implementation. SimpleSAMLphp IdP and SP code-bases have been modified to include the AttributeQuery statement, containing the list of required attributes, with the SAML AuthnRequest statement at the SP side as well as to parse the list of attributes from the AttributeQuery statement of the authentication request at the IdP side and then to show them at the top of the consent form (Figure 4). The modified SAML authentication request containing i number of requested attributes can be modelled in this way:

$$AuthnReq' = (id_{req}, id_{sp}, (a_1, a_2, ...a_i)).$$

We believe that the consent form is the best place to show the required attributes as it will allow a user to choose an IdP from where she can aggregate attributes as well as to decide the minimum number of attributes she needs to release to an SP. In addition, when a user chooses another external SAML IdP for attribute aggregation, these requested attributes will also be included within the AttributeQuery element of the SAML AuthnRequest that will be sent to the external IdP. This will enable any external SAML IdP to parse and show the requested attributes at any appropriate place. Moreover, the capability has been added to the SP code-base that will allow the admin to set the required attributes for each SP in the configuration file (called *config.php* in SimpleSAMLphp) which will be used to configure the AttributeQuery element which ultimately is embedded inside the AuthnRequest statement as described before. In this manner, the admin can set different attribute requirements for different SPs.



The attributes requested by the https://192.168.1.85/simplesaml/module.php/saml/sp/metadata.php/default-sp are: telephone, age, position, org.

Information that will be sent to https://192.168.1.85/simplesaml/module.php/saml/sp/metadata.php/default-sp

Attributes from IdP:https://192.168.1.115/simplesaml/saml2/idp/metadata.php:

□username:ripul

□name:Ripul Test

□email:ripul@er.et

□telephone:01234445566

□age:34

Fig. 4: The requested attributes at the consent form.

The current implementation equipped with all these features can aggregate attributes from different SAML IdPs. The implementation supports two types of SAML IdPs: trusted and semi-trusted. A trusted SAML IdP is the one which has been federated in the traditional way by exchanging metadata at the admin level whereas a semi-trusted SAML IdP is the one that has been federated using the concept of dynamic federation (Ferdous & Poet, 2013b). In addition, the Hybrid IdP determines the LoA for each IdP in this way: attributes from the Hybrid IdP or from any trusted SAML IdP will have a LoA value of 2 whereas attributes from all other IdPs will have a LoA value of 1. The LoA level signifies the confidence the Hybrid IdP has on those IdPs.

At this point, we would like to compare our proposed approach with an already implemented attribute aggregation method. In (D. Chadwick & Inman, 2013), authors have presented an approach for attribute aggregation from SAML IdPs. The approach is loosely based on the Identity Proxying model where the the role of a Proxy IdP has been replaced with a web service called Trusted Attribute Aggregation Service or TAAS. To access any service from an SP, a user is forwarded to the TAAS which aggregates attributes from different IdPs and sends the aggregated set of attributes to the SP. As such, this is similar to the Identity Proxying capability of the Hybrid IdP. However, there are several distinct differences between our model and the TAAS. Firstly, TAAS does not have the Identity Relay capability meaning that all attributes released by other IdPs are exposed to TAAS. Secondly, TAAS requires every user to install a browser plugin to initate the protocol flow whereas our model does not rely on any such external component.

5 Use-cases

In this section we present two use-cases to show the applicability of our model. Each use-case illustrates the interaction between different entities utilising the implemented Hybrid model to aggregate attributes from multiple SAML IdPs. To present each interaction we use the following notation:

$$e_1 \xrightarrow{HTTPS} e_2 : \text{msg/resource}$$

The notation represents that a message (request/response) or a resource is transferred from entity e_1 to entity e_2 using a secure HTTPS channel. Moreover, we denote a user with u, an SP with sp, the Hybrid IdP as $hy-idp$ and other IdPs with idp_i where $i = 1...n$.

Each use-case is presented below.

5.1 Use-case 1

The first use-case illustrates the simplest form of attribute aggregation using the Hybrid model. The initial setup is: a user wants to access a service from an SP. The SP requires a set of user attributes from the IdP(s). The SP is deployed using our modified SimpleSAMLphp and is federated with the Hybrid IdP. The other SAML IdP is also implemented using SimpleSAMLphp. Moreover, the admin of the SP has configured the required attributes for each service in the configuration file as mentioned above. With this setup, the protocol flow for the use-case is illustrated in Figure 5 and is described below:

1. The user visits the SP to access one of its services. The user submits an access request for the service.

2. The user is forwarded to the WAYF Page of the SP to choose the IdP.

3. The user chooses the Hybrid IdP.

4. The required attribute(s) for the requested service are read from the configuration file and are used to create a SAML authentication request. The user is redirected to the Hybrid IdP with the SAML authentication request ($AuthnReq'$).

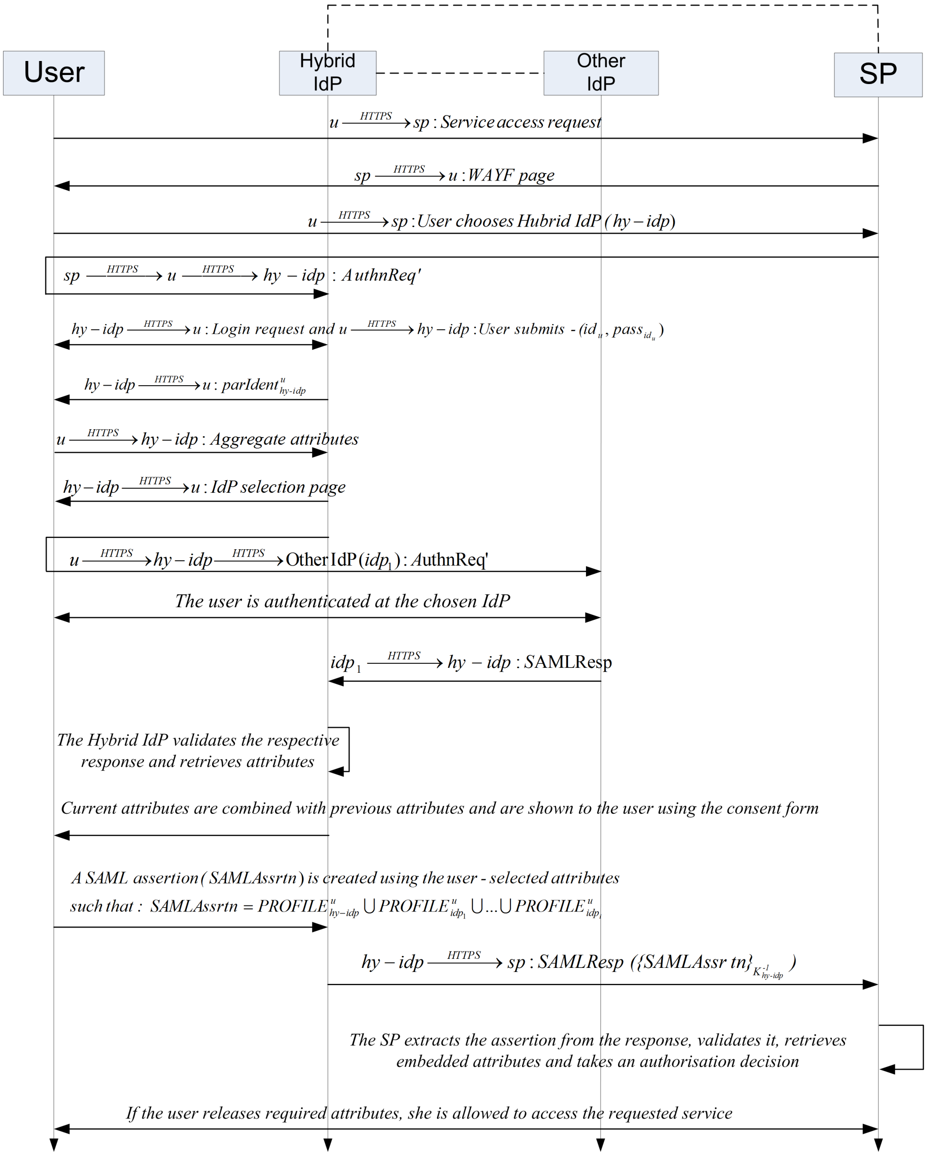

User
Hybrid IdP
Other IdP
SP
$u \xrightarrow{HTTPS} sp$: Service access request
$sp \xrightarrow{HTTPS} u$: WAYF page
$u \xrightarrow{HTTPS} sp$: User chooses Hubrid IdP ($hy-idp$)
$sp \xrightarrow{HTTPS} u \xrightarrow{HTTPS} hy-idp$: AuthnReq'
$hy-idp \xrightarrow{HTTPS} u$: Login request and $u \xrightarrow{HTTPS} hy-idp$: User submits - (id_u , $pass_{id_u}$)
$hy-idp \xrightarrow{HTTPS} u$: $parIdent^{u}_{hy\text{-}idp}$
$u \xrightarrow{HTTPS} hy-idp$: Aggregate attributes
$hy-idp \xrightarrow{HTTPS} u$: IdP selection page
$u \xrightarrow{HTTPS} hy-idp \xrightarrow{HTTPS}$ Other IdP (idp_1) : AuthnReq'
The user is authenticated at the chosen IdP
$idp_1 \xrightarrow{HTTPS} hy-idp$: SAMLResp
The Hybrid IdP validates the respective response and retrieves attributes
Current attributes are combined with previous attributes and are shown to the user using the consent form
A SAML assertion (SAMLAssrtn) is created using the user - selected attributes
such that : $SAMLAssrtn = PROFILE^{u}_{hy-idp} \cup PROFILE^{u}_{idp_1} \cup ... \cup PROFILE^{u}_{idp_i}$
$hy-idp \xrightarrow{HTTPS} sp$: SAMLResp ($\{SAMLAssrtn\}_{K^{-1}_{hy\text{-}idp}}$)
The SP extracts the assertion from the response, validates it, retrieves embedded attributes and takes an authorisation decision
If the user releases required attributes, she is allowed to access the requested service

Fig. 5: Protocol flow for Use-case 1.

5. The user is authenticated at the Hybrid IdP after providing his identifier (user-id) and credential (password).

6. Once the user is authenticated, the consent form is shown which consists of the user attributes and their respective values (the partial identity, $parIdent^{u}_{hy-idp}$) stored in the Hybrid IdP.

7. The consent form displays the requested attributes from the SP (Figure 4). In addition, the consent form allows the user to aggregate more attributes from other IdPs as well (Figure

2). The user compares the requested attributes and the attributes shown in the consent form to decide if she needs to aggregate attributes.

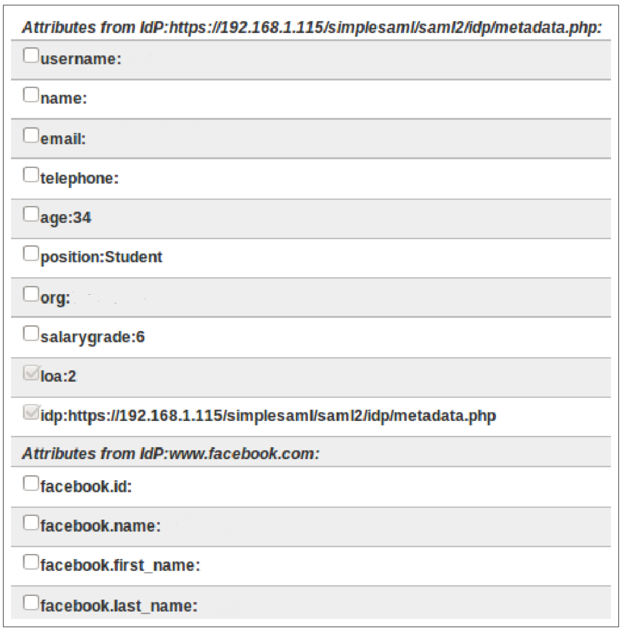
8. Assuming the user decides to aggregate attributes, she clicks the *Aggregate More Attributes* button.

9. A session is created to keep track of the previous attributes and then the user is forwarded to the IdP selection page of the Hybrid IdP. The page contains the list of only those IdPs that have not been used already.

10. The user chooses one IdP (*idp*$_1$) and based on the user's selection, the respective protocol is initiated. For example, a SAML authentication request is created and the user is forwarded to the respective IdP.

11. The user authenticates at the respective IdP and releases the attributes to the Hybrid IdP using their respective protocol and implementations. For example, if the user chooses a SAML IdP that supports selective disclosure (e.g. SimpleSAMLphp), the user is shown a basic consent form where she chooses the attributes. Based on her selection, a SAML response will be returned to the Hybrid IdP.

12. Once the user returns to the Hybrid IdP, it validates the SAML assertion using its corresponding mechanism. Then, the attributes are retrieved from the assertion. Next, the previously aggregated attributes are retrieved using the saved session and these two sets of attributes are then merged together using the data structure mentioned previously and are shown to the user using the consent form (Figure 6). From the figure, the attributes are grouped based on the IdP. The Hybrid IdP determines the LoA for each IdP. Note that, the IdP attribute and the LoA attribute for each attribute group are disabled so that the user cannot unselect them. If the user chooses to release at least one attribute from an attribute group, the respective IdP and the LoA value will be attached along with the chosen attribute(s).



Attributes from IdP:https://192.168.1.115/simplesaml/saml2/idp/metadata.php:

□username:

□name:

□email:

□telephone:

□age:34

□position:Student

□org:

□salarygrade:6

☑loa:2

☑idp:https://192.168.1.115/simplesaml/saml2/idp/metadata.php

Attributes from IdP:www.facebook.com:

□facebook.id:

□facebook.name:

□facebook.first_name:

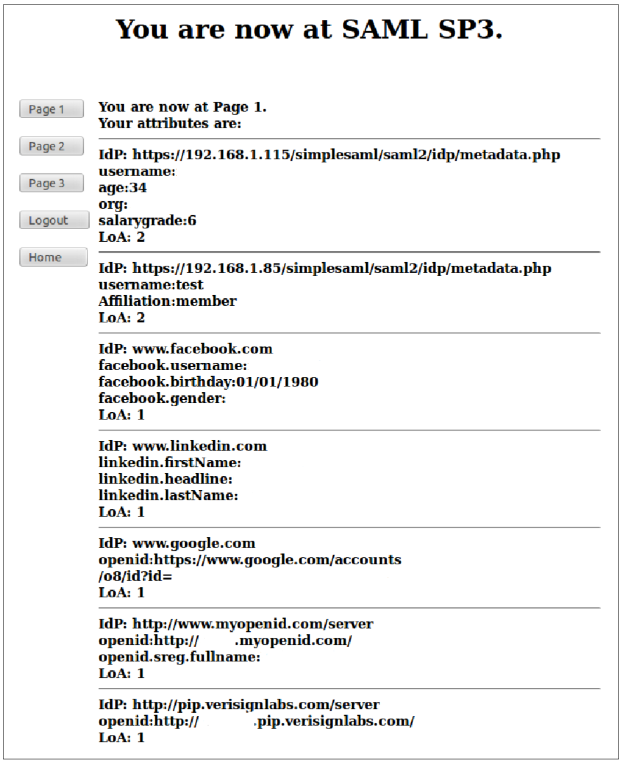
□facebook.last_name:

Fig. 6: Aggregated attributes from two IdPs at the consent form.

13. The requested attributes from the SP are listed at the consent form. The user can always consult that list to determine if she needs to aggregate more attributes. If so, the steps 4-7 are repeated.

14. Once the user has aggregated the required attributes, she selects those attributes and based on the user selection, attributes from each group are wrapped inside a SAML AttributeStatement element and then different such elements are wrapped inside a SAML AttributeStatements element. Then a SAML Response is created with the AttributeStatements element which is wrapped inside a SAML assertion and is sent back to the SP.

15. Upon receiving the assertion, the SP validates the assertion and retrieves the SAML AttributeStatements element. From there each attribute group is retrieved and from each group, all attributes are extracted. A sample of such attributes is illustrated in Figure 7. Then, the SP can take authorisation decisions based on those attributes which is not explored any further.



You are now at SAML SP3.

```
You are now at Page 1.
Your attributes are:

IdP: https://192.168.1.115/simplesaml/saml2/idp/metadata.php
username:
age:34
org:
salarygrade:6
LoA: 2

IdP: https://192.168.1.85/simplesaml/saml2/idp/metadata.php
username:test
Affiliation:member
LoA: 2

IdP: www.facebook.com
facebook.username:
facebook.birthday:01/01/1980
facebook.gender:
LoA: 1

IdP: www.linkedin.com
linkedin.firstName:
linkedin.headline:
linkedin.lastName:
LoA: 1

IdP: www.google.com
openid:https://www.google.com/accounts
/o8/id?id=
LoA: 1

IdP: http://www.myopenid.com/server
openid:http://       .myopenid.com/
openid.sreg.fullname:
LoA: 1

IdP: http://pip.verisignlabs.com/server
openid:http://          .pip.verisignlabs.com/
LoA: 1
```

Fig. 7: Released attributes from multiple IdPs in a single SP session.

5.2 Use-case 2

The second use-case illustrates the scenario when the Hybrid IdP acts as the Relay IdP using the IR model to ensure that another SAML IdP can release encrypted assertions to an SP as discussed previously. The setup is very similar to the first use-case except that the other SAML IdP needs to be federated with the SP to allow the other IdP to create an encrypted assertion targeted for the SP. We are also assuming that the user has already gone through the steps from 1 - 7 from the previous flow. With this setup, the protocol flow for the use-case is illustrated in Figure 8 and is described below:

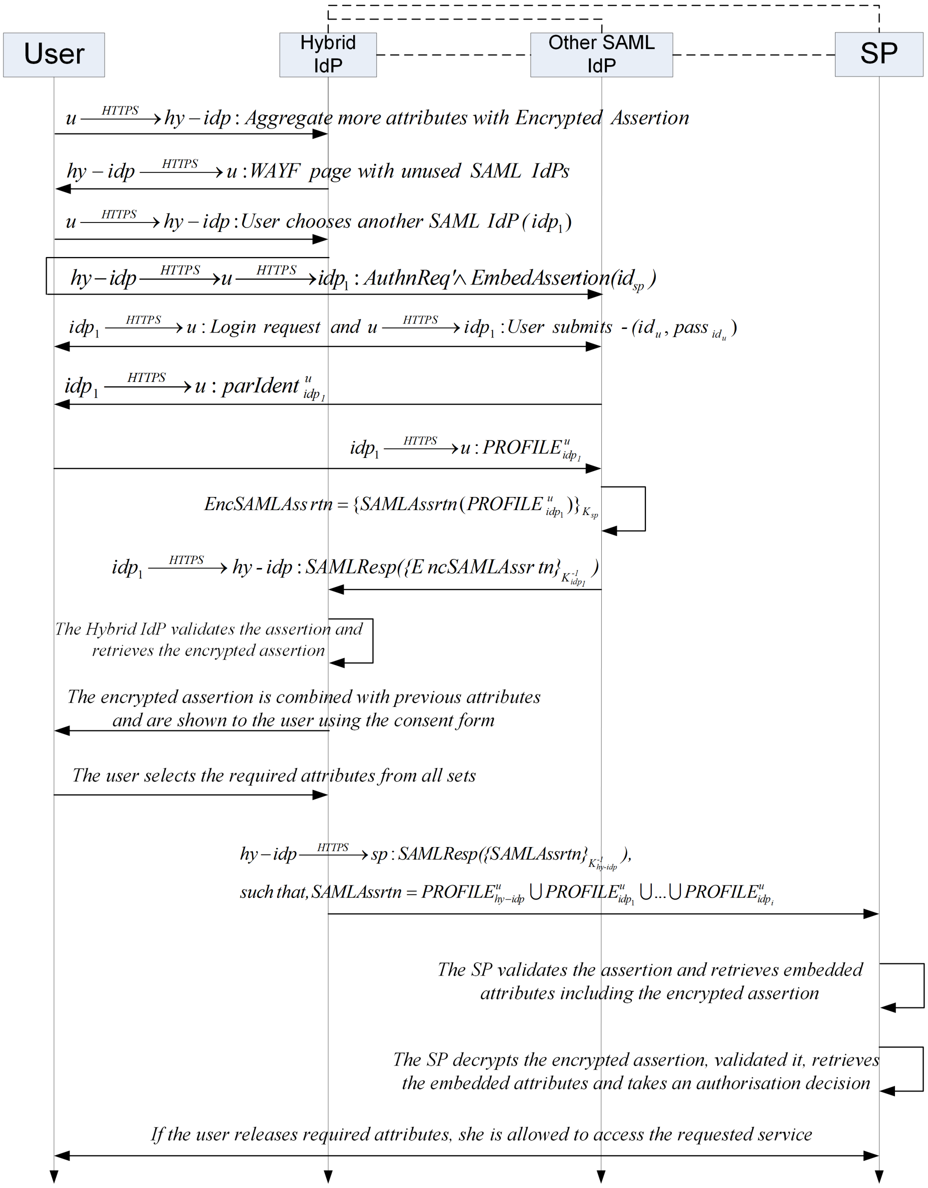

User
Hybrid IdP
Other SAML IdP
SP
$u \xrightarrow{HTTPS} hy-idp$: Aggregate more attributes with Encrypted Assertion
$hy-idp \xrightarrow{HTTPS} u$: WAYF page with unused SAML IdPs
$u \xrightarrow{HTTPS} hy-idp$: User chooses another SAML IdP (idp_1)
$hy-idp \xrightarrow{HTTPS} u \xrightarrow{HTTPS} idp_1 : AuthnReq' \wedge EmbedAssertion(id_{sp})$
$idp_1 \xrightarrow{HTTPS} u$: Login request and $u \xrightarrow{HTTPS} idp_1$: User submits - $(id_u, pass_{id_u})$
$idp_1 \xrightarrow{HTTPS} u : parIdent^u_{idp_1}$
$idp_1 \xrightarrow{HTTPS} u : PROFILE^u_{idp_1}$
$EncSAMLAssrtn = \{SAMLAssrtn(PROFILE^u_{idp_1})\}_{K_{sp}}$
$idp_1 \xrightarrow{HTTPS} hy\text{-}idp : SAMLResp(\{EncSAMLAssrtn\}_{K^{-1}_{idp_1}})$
The Hybrid IdP validates the assertion and retrieves the encrypted assertion
The encrypted assertion is combined with previous attributes and are shown to the user using the consent form
The user selects the required attributes from all sets
$hy-idp \xrightarrow{HTTPS} sp : SAMLResp(\{SAMLAssrtn\}_{K^{-1}_{hy-idp}})$,
such that, $SAMLAssrtn = PROFILE^u_{hy-idp} \cup PROFILE^u_{idp_1} \cup ... \cup PROFILE^u_{idp_i}$
The SP validates the assertion and retrieves embedded attributes including the encrypted assertion
The SP decrypts the encrypted assertion, validated it, retrieves the embedded attributes and takes an authorisation decision
If the user releases required attributes, she is allowed to access the requested service

Fig. 8: Protocol flow for Use-case 2.

1. The user needs to aggregate attributes from a SAML IdP and she does not want to reveal the attributes to the Hybrid IdP. Therefore, she selects the *Return Encrypted Assertion* checkbox and clicks the *Aggregate More Attributes* button.

2. A session is created to keep track of the previous attributes and then the user is forwarded to the IdP selection page of the Hybrid IdP just like before. However, this time the page enlists only those SAML IdPs from which attributes have not been aggregated yet.

2. When the user chooses one SAML IdP (idp_1), a SAML authentication request is created. An attribute called *EmbedAssertion* in inserted within the request containing the Entity ID of the SP. Then the user is forwarded to the SAML IdP with the request.

3. The IdP notes the *EmbedAssertion* attribute in the request and the entity ID of the SP is stored in a session variable.

4. The user is authenticated at the IdP and then the consent form containing attributes (the partial identity of the user) stored at this IdP (idp_1) is shown.

5. The user chooses the attributes and clicks the *Release attributes to SP* button.

6. The IdP uses the entity ID from the session variable to create an encrypted assertion ($\{EncSAMLAssrtn\}_{K_{idp_1}^{-1}}$) targeted for the SP using the chosen attributes.

7. The encrypted assertion is then Base64 encoded and added as the value of the *encryptedAssertion* attribute and is used to create a regular unencrypted SAML assertion. The regular assertion is then sent back to the Hybrid IdP.

8. Upon receiving this assertion, the Hybrid IdP validates the assertion and retrieves the *encryptedAssertion* attribute. Since this encrypted assertion is not decryptable by the Hybrid IdP, the IdP treats it as it is.

9. Then, the Hybrid IdP merges the encrypted assertion with the previously aggregated attributes using the approach described before and displays all of them in different attribute groups in the consent page. Since, the Base64-encoded encrypted assertion can be quite a large value, it is not shown in the consent page. Instead, the text *Encrypted Assertion from the Other IdP* is shown (Figure 9). Note that the Hybrid IdP assigns a LoA value of 2 for the encrypted assertion as the other SAML is assumed to be trusted in this instance.

10. When the user chooses the encrypted assertion (along with other attributes from other IdPs) and clicks the *Release attributes to SP* button, an assertion containing these attributes is created as described previously and is sent back to the SP.

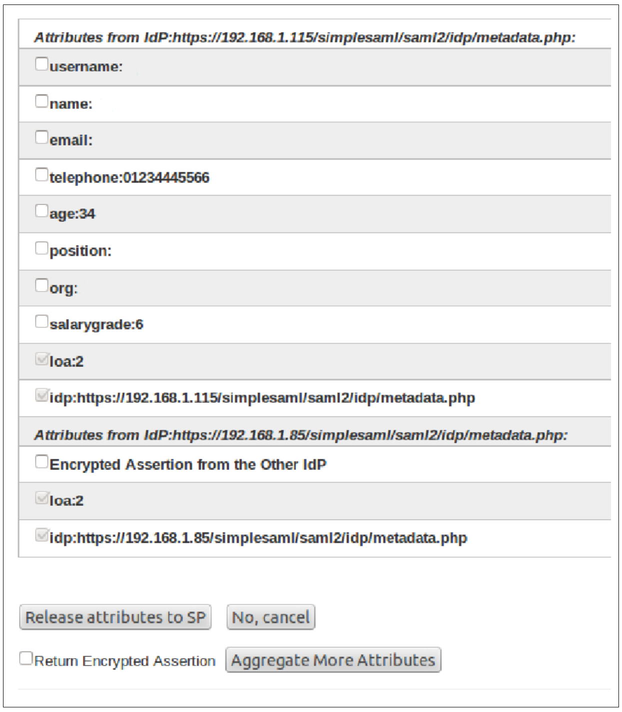
11. The SP validates the assertion and retrieves attributes as discussed previously. However, when the SP finds the *encryptedAssertion* attribute, it is treated in a special way. The attribute is, at first, Base64-decoded. Then the SP decrypts the assertion, validates it and retrieves the attributes. Then the SP can take an authorisation decision as described above.

6 Discussion

In this section, we will analyse if our proposed model can satisfy all the required functional, security and privacy requirements. We will also describe the advantages and the limitation of our model. In addition, possible future work will be discussed.

6.1 Analysis

The two use-cases illustrated previously require that the Hybrid IdP and the SP are part of the same federation as well as other SAML IdPs and the PPIdP are federated with the SP thereby satisfying *F1*. The Hybrid IdP also retains the sessions while the user is forwarded to aggregate attributes and performs a dual role as an IdP to the SP and as an SP to other IdPs thereby satisfying *F2* and *F3*. When other SAML IdPs return assertions to the Hybrid IdP, it validates the assertions as required by the SAML protocol and extracts the attributes and thus satisfies *F4*. Moreover, the approach of returning encrypted assertions to the Hybrid IdP satisfies *F5*. As evident from the use-cases, the Hybrid IdP filters out an IdP once it has been used for attribute aggregation and hence satisfies *F6*. The Hybrid IdP groups attributes



Attributes from IdP:https://192.168.1.115/simplesaml/saml2/idp/metadata.php:

□username:

□name:

□email:

□telephone:01234445566

□age:34

□position:

□org:

□salarygrade:6

☑loa:2

☑idp:https://192.168.1.115/simplesaml/saml2/idp/metadata.php

Attributes from IdP:https://192.168.1.85/simplesaml/saml2/idp/metadata.php:

□Encrypted Assertion from the Other IdP

☑loa:2

☑idp:https://192.168.1.85/simplesaml/saml2/idp/metadata.php

Fig. 9: Encrypted assertion as the attribute in the consent page.

according to their sources using the data structure described in Listing 2 with the proper LoA value and the SP has been equipped with the capability to interpret such groupings, therefore satisfying *F7* and *F8*. A user can easily switch back and forth between the IP and the IR mode by just selecting a checkbox. Depending on a user's selection, the Hybrid IdP returns aggregated attributes or assertions. This satisfies *F9*.

The security requirement *S1* is mostly dependant on the respective IdP. To prove its trustworthiness to its users, an IdP should deploy mechanisms that will enable secure registration and authentication. For our implementation, users can only access their attributes after being authenticated and the authentication takes place over secure HTTPS ($\xrightarrow{HTTPS}$) channel. Hence, *S1* is fulfilled. Security requirements *S2*, *S3* and *S4* can be satisfied using cryptographic mechanisms. The SAML implementations (including ours) transmit user attributes over secure HTTPS ($\xrightarrow{HTTPS}$) channel which ensures that data during transmission are not disclosed to any party, thereby satisfying *S2*. Furthermore, the IR model can be used for SAML IdPs to release an encrypted SAML assertion (*EncSAMLAssrtn*) which is encrypted using the public key of the SP ($\{SAMLAssrtn\}_{K_{sp}}$) to enure that released attributes from a SAML IdP is not even disclosed to the Hybrid IdP. Similarly, these three standards use digital signature (e.g. $\{SAMLAssrtn\}_{K_{idp}^{-1}}$, the digitally signed SAML assertion) to ensure that data during transmission are not altered without being noticed and to enable non-repudiation. This satisfies *S3* and *S4*. Moreover, the Hybrid IdP itself does not store any released attributes and it determines and assigns the LoA in a correct manner and thus satisfies *S5* and *S6*.

The Hybrid IdP releases SP-specific pseudonymous identifiers to different SPs. This undermines the possibility of malicious SPs colluding together to track a user across multiple SPs and build a profile. Thus, it fulfils *P1*. The Hybrid IdP has support for the selective disclosure of attributes which enables users to select the attributes that they want to release to an SP. For other IdPs, how this property is fulfilled will depend on the respective IdP. As discussed before, Shibboleth SAML implementation does not have any support for the selective disclosure of attributes. In such scenarios, the Hybrid IdP provides another layer using the consent module to select attributes before they are released to an SP and hence it satisfies *P2*. Moreover, no attributes from the PPIdP are automatically released to an SP and the selective disclosure of attributes using the consent module ensures that users provide their explicit consent before they are released. This fulfils *P3*. Our implementation also enables an SP to inform a user which attributes are required to access a service and this ensures data minimisation as the user does not need to release all attributes to an SP. This satisfies *P4*. The SAML protocol does not have any mechanism for remote administration of policies that could be used to enforce *P5* and also there is not a single SAML implementation that allows this features. We plan to work on this in future.

6.2 Advantages

The hybrid model offers a number of advantages which we describe below:

- By combining the features of the IP and IR models, we have been able to leverage the advantages of both models which allow a user to aggregate attributes almost seamlessly. The previously implemented models such as the LS and the IFL model require a user to engage in additional steps (e.g. the linking procedure) whereas the proposed approach does not require any such additional steps.
- Our model allows a user to switch between the IP and IR mode in the most convenient way - just by selecting a checkbox. Then all mechanisms are handled internally without burdening the user.
- Another added advantage that comes from the SSO feature of the SAML is that once the Hybrid IdP releases attributes to an SP, the user does not need to log in and choose attributes for the same SP until the session is lost. However, when the user accesses services from another SP and chooses the Hybrid IdP, the SSO feature directly takes the user to the consent form containing the already aggregated attributes from the first instance with the first SP. At this point, the user can aggregate more attributes if she wishes or chooses a separate set of attributes. In this way, the user can release completely different sets of attributes to different SPs using either the same or extended sets of aggregated attributes.
- None of the current SAML implementations allows an SP to pass on information regarding requested attributes along with the authentication request. The approach we have adopted to pass such request via the SAML authentication request is the first of its kind and could be widely adopted not only for the attribute aggregation mechanism but also for any general SAML implementation.

6.3 Limitations

There are a few limitations with our current implementation. These limitations are discussed below:

The LoAs of different IdPs are currently hard-coded. The way we have assigned LoAs might not suit every scenario. The best approach would be to allow the admin to assign the LoA at the configuration file and then read them during run-time.

There is another limitation when a PPIdP is used for aggregating attributes: the PPIdP allows a user to access the PPIdP only from a browser in the mobile phone where it is installed. This is not a realistic assumption for wide-scale adaptation of the approach.

Even though our approach allows any user to aggregate attributes from multiple SAML IdPs in single a session considering different security and privacy issues, the associated burden during the process from the perspective of a user needs to be considered. A user needs to go through several rounds of selection process - selecting the Hybrid IdP and then selecting a number of required IdPs and choosing required attributes from each of them - which might be a daunting experience for many users. For a successful deployment of the proposed approach, it must be justifiable to a user and the associated burden must be outweighed by its advantages. This will largely depend on the use-case scenarios for which the Hybrid model is deployed. This calls for a usability study for different scenarios. Depending of the result of that study, appropriate modifications may be suggested.

Furthermore, our approach assumes an SP to be honest in asking the number of attributes required for accessing its services. A malicious SP may request more than required attributes and once these attributes are received, they may be abused. This assumption finds it bases on the current setting of federated services in which an SP is never questioned on why it requires specified attributes for any specific services. One way to tackle this issue is to enforce privacy requirement *P5*. This will empower a user to understand how her attributes are utilised in an SP and help her to decide if the SP has been honest in requesting attributes. Furthermore, a dishonest SP can launch a phishing attack by redirecting users to a fake Hybrid IdP. From there, users can be redirected to fake IdPs which may have the same look and feel of the original IdPs. This would trick any user to expose their identifier and credential to the adversary. The authenticity of the Hybrid IdP can be verified using digital certificates with web PKI (Public Key Infrastructure) which can undermine this attack. Unfortunately, many general users still do not understand the technical details of this technology and hence are not proficient in verifying the authenticity of a website using digital certificates. This leaves them prone to such attacks.

6.4 Future Work

There are several directions to take from here:

- One major problem that still exists almost in every SAML system is the lack of control over the attributes once they are released to the SP. One way to achieve any control in this regard could be the use of remote administration of policies between two entities. None of the SAML implementation currently supports this mechanism. It will be interesting to investigate how such mechanisms can be integrated within the SAML.
- We are also working on how to allow the admin to configure LoAs for different IdPs at the configuration file.
- With the proliferation of online services and social networks, more and more user attributes will be stored online across multiple IdPs, most of which are non-SAML IdPs. Therefore, a mechanism to aggregate and manage those attributes from a central place will be useful for the users. We are investigating how the Hybrid model can be extended to aggregate attributes from non-SAML IdPs.

7 Conclusion

With the proliferation of online services, more and more user attributes will be stored online across multiple IdPs. Innovative service scenarios in the near future would definitely need a mechanism to allow users to aggregate attributes from multiple sources in the simplest of ways. The existing models are complex and require preliminary steps which are not intuitive.

In this paper we present a hybrid way of aggregating attributes from differnt SAML IdPs in a single service session. The approach, based on the Identity Proxying and Relay model, is simple and does not require any preliminary step or complex user interactions like previous existing methods. Depending on the requirement, the user can switch back and forth between the proxy and relay model by selecting or un-selecting a single checkbox. Unlike any previous model, it allows the proxy and/or Relay IdP to retain the source of the aggregated attributes which would be beneficial for an SP to take any authorisation decision. We have proposed a few additional modifications of the SAML protocol and a novel way of letting an IdP know the attribute requirements of an SP which is missing in the current SAML specification. We strongly believe that our approach has true potential to advance federated service scenarios to the next step and to begin a new era of next generation federated services.

References

Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen. (2011). *Virtual collaboration attribute management.* Accessed on 1 May, 2013. (http://www.surfnet.nl/nl/Innovatieprogramma's/gigaport3/Documents/EDS%2011-06%20AttributeManagement%20v1.0.pdf)

Cantor, S. (7 January, 2008). *Shibboleth Attribute Release Policies.* (https://wiki.shibboleth.net/confluence/display/SHIB/IdPARPConfig)

Chadwick, D., & Inman, G. (2009). Attribute aggregation in federated identity management. *Computer*, *42*(5), 33–40.

Chadwick, D., & Inman, G. (2013, Sept). The Trusted Attribute Aggregation Service (TAAS) - Providing an Attribute Aggregation Layer for Federated Identity Management. In *Eighth International Conference on Availability, Reliability and Security (ARES), 2013* (p. 285=-290). doi: 10.1109/ARES.2013.38

Chadwick, D. W. (2009, January). Federated Identity Management. In A. Aldini, G. Barthe, & R. Gorrieri (Eds.), *FOSAD 2008/2009* (pp. 96–120). Berlin: Springer-Verlag. Retrieved from http://www.cs.kent.ac.uk/pubs/2009/3030

Chadwick, D. W., Inman, G. L., Siu, K. W., & Ferdous, M. S. (2011). Leveraging social networks to gain access to organisational resources. In *Proceedings of the 7th ACM workshop on Digital identity management* (pp. 43–52). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2046642.2046653 doi: 10.1145/2046642.2046653

Chappell, D. (2006, April). *Introducing Windows CardSpace.* (http://msdn.microsoft.com/en-us/library/aa480189.aspx)

De Cock, D., Wouters, K., Schellekens, D., Singelee, D., & Preneel, B. (2005). Threat modelling for security tokens in web applications. In *Communications and multimedia security* (pp. 183–193).

Desmet, L., Jacobs, B., Piessens, F., & Joosen, W. (2005). Threat modelling for web services based web applications. In *Communications and multimedia security* (pp. 131–144).

Dominicini, C. K., Simplício Jr, M. A., Sakuragui, R. R., Carvalho, T. C., Näslund, M., & Pourzandi, M. (2010). Threat modeling an identity management system for mobile internet. *Rio de Janeiro, Brasil.* (http://www.teses.usp.br/teses/disponiveis/3/3141/tde-23032012-101827/publico/Tese_RonySakuragui.pdf)

Ferdous, M. S., Chowdhury, M. J. M., Moniruzzaman, M., & Chowdhury, F. (2012, May). Identity federations: A new perspective for Bangladesh. In *Informatics, Electronics Vision (ICIEV), 2012 International Conference on* (pp. 219–224).

Ferdous, M. S., Jøsang, A., Singh, K., & Borgaonkar, R. (2009). Security Usability of

Petname Systems. In A. Jøsang, T. Maseng, & S. Knapskog (Eds.), *Identity and Privacy in the Internet Age* (Vol. 5838, pp. 44–59). Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-04766-4_4

Ferdous, M. S., Norman, G., & Poet, R. (2014). Mathematical modelling of identity, identity management and other related topics. In *Proceedings of the 7th international conference on security of information and networks* (p. 9).

Ferdous, M. S., & Poet, R. (2013a). Analysing Attribute Aggregation Models in Federated Identity Management. In *Proceedings of the 6th International Conference on Security of Information and Networks* (pp. 181–188). ACM.

Ferdous, M. S., & Poet, R. (2013b). Dynamic Identity Federation Using Security Assertion Markup Language (SAML). In S. Fischer-Hübner, E. Leeuw, & C. Mitchell (Eds.), *Policies and Research in Identity Management* (Vol. 396, pp. 131–146). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-37282-7_13 doi: 10.1007/978-3-642-37282-7_13

Jøsang, A., Al, M., & Suriadi, Z. S. (2007). Usability and privacy in identity management architectures. In *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers'* (pp. 143–152).

Khattak, Z. A., Sulaiman, S., & Manan, J. (2010). A study on threat model for federated identities in federated identity management system. In *Information technology (itsim), 2010 international symposium in* (Vol. 2, pp. 618–623).

Klingenstein, N. (2007). Attribute Aggregation and Federated Identity. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on* (pp. 26–26).

Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Threat modeling as a basis for security requirements. In *Symposium on requirements engineering for information security (sreis)* (Vol. 2005, pp. 1–8).

NISTWP. (2006, April). *Electronic Authentication Guideline: INFORMATION SECURITY.* (http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf)

OpenID Authentication 2.0 - Final. (2007). 5 December. (http://openid.net/specs/openid-authentication-2_0.html)

Sampo Kellomäki. (2008). *Query Extension for SAML AuthnRequest (Draft).* 22 April. (http://zxid.org/tas3/anrq-index.html)

Shibboleth. (2016). (http://www.internet2.edu/products-services/trust-identity/shibboleth/)

SimpleSAMLphp. (2016). (http://simplesamlphp.org/)

Standard, O. (2005). *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.* 15 March. (http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)

ZXID. (2016). (http://www.zxid.org/)