Analysing Attribute Aggregation Models in Federated Identity Management

Md. Sadek Ferdous
School of Computing Science, University of Glasgow
Glasgow, G12 8QQ, Scotland
m.ferdous.1@research.gla.ac.uk

Ron Poet
School of Computing Science, University of Glasgow
Glasgow, G12 8QQ, Scotland
ron.poet@glasgow.ac.uk

ABSTRACT

This paper presents a comparative analysis of different attribute aggregation models against a set of requirements in the settings of the Federated Identity Management (FIM). There are several attribute aggregation models currently available which allow the user to collate attributes from multiple identity providers (IdP in short) in a single service. These models impose different novel requirements which have never been analysed before and there lacks a thorough analysis of these models that will compare them side-by-side against a set of requirements. We aim to fill in these gaps in this work. We have formulated a set of trust, functional, security and privacy requirements that are needed for each model and shown the interlink between these requirements. These requirements have been used to compare the models side-by-side in tabular forms which would allow the readers to instantly identify the requirements for each model, the advantages it offers and the weaknesses it has.

Categories and Subject Descriptors

C.2.0 [**General**]: [Security and protection]; D.4.6 [**Security and Protection**]: [Access controls, Authentication]

General Terms

Design, Security

Keywords

Attribute Aggregation, Identity Management, Federated Identity Management, Trust.

1. INTRODUCTION

We have experienced a tremendous popularity and expansion of online services in last 15 years or so. Many of these services require that the user must prove their identities before accessing the services. The term *Digital Identity*, a projection of one's identity consisting of different attributes encoded in electronic formats [4], is used in such scenarios. Since, different services require different types of attributes, the users end up with different digital identities stored at different providers. These separated digital identities are known as the Partial Digital Identity (or Partial Identity, in short) of the user.

Identity Management (IdM, in short) was introduced by the industry to facilitate online management of user identities (digital identities of the user) which resulted in various different Identity Management Systems (IMS, in short). Shibboleth [1], OpenID [3], Microsoft's CardSpace [8], etc. are all examples of different IMS. Each IMS has several parties involved which are: Service Provider (SP) or Relying Party (RP) - an entity that provides services to the clients or to the other service providers, Identity Provider (IdP) - an entity that holds and releases the partial identity (user attributes) to enable the user to receive services from a SP and Client/User - an entity that receives services from a SP.

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). A federation with respect to the Identity Management is a business model in which a group of two or more trusted parties legally bind themselves with a business and technical contract [6, 14]. The IdPs and SPs who bind themselves in such a way form the so-called Circle of Trust (CoT) which make 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 system and then access services from federated service providers without further logins. Security Assertion Markup Language (SAML) [15] and OpenID [3] are two of the most popular identity federation technologies. In this paper, our main focus is on the SAML-based identity federation. Shibboleth [1] and SimpleSAMLphp [2] are currently two popular implementations based on SAML. To access a service from a SP in SAML, the user is forwarded to the chosen IdP where the user authenticates. Then, the user is sent back to the SP with an assertion that contains user attributes. The SP validates the assertion, retrieves attributes from it and based on these attributes, the SP decides if the user can access the service.

Even though the federated services improve the usability and experience of online services, there exists a serious limitation: the user is allowed to chose only one partial identity from one single IdP during one single service session at a SP [7]. The concept of Attribute Aggregation has been introduced to tackle this very problem that will allow the user to aggregate attributes from multiple IdPs in a single service

SIN '13, November 26 - 28 2013, Aksaray, Turkey
Copyright 2013 ACM 978-1-4503-2498-4/13/11...$15.00.
http://dx.doi.org/10.1145/2523514.2526998.

To appear in the proceedings of the 6th International Conference on Security of Information and Networks (SIN'13).

session. A number of novel approaches exist in reality for aggregating attributes, each with their own strengths and weaknesses. Each of these methods, depicted in different models, has additional interactions, require additional entities to function and impose novel requirements. The existing literature of attribute aggregation discuss these methods and their interactions in a casual way without analysing these requirements [12, 5, 10]. These make it difficult for anyone not only to have a thorough understanding on these models, but also to compare them against a set of requirements. We aim to address this issue in this paper where we compare each model side-by-side against a set of requirements and analyse their strengths and weaknesses in a more systematic way. The contributions of this paper are:

- At first, we have created a taxonomy of existing attribute aggregation models.
- We have formulated the types of requirements and then populated each type with different requirements for the basic Federated Identity Management.
- Then, we have treated each model individually to analyse the novel requirements it has to consider in addition to the basic set of requirements.
- We have shown the interlink between different requirement that illustrates how meeting one or more requirements leads to satisfying another requirement.
- Finally, we have presented our findings in tabular forms making it easier for the reader to instantly identify the requirements for each model, the advantages it offers and the weaknesses it has.

With this introduction, this paper is organised as follows. We formulate a basic set of Functional, Trust, Security and Privacy requirements that are needed for the traditional federated services in Section 2. Then, we define the taxonomy of different Attribute Aggregation Models in Section 3. We have analysed each model separately to devise the additional requirements that a single model imposes in Section 4, 5 and 6. We present our findings in tabular forms in Section 7 and some existing works on Attribute Aggregation are discussed in Section 8. Finally, we conclude in Section 9.

2. A STUDY OF REQUIREMENTS

There are different types of requirements for the Federated Identity Management. We classify them as four types: Functional, Trust, Security and Privacy. Each of them is analysed below.

Functional Requirements (FR): The functional requirement outlines the way federations must be created between the SP, whose services the user is trying to access, and the IdPs, from where attributes will be aggregated, for each model and any other requirements that are needed to ensure its functionalities. The type of federations that will need to be created will vary from one model to another. For the traditional simple FIM model with a single CoT where there are one or more IdPs and one or more SPs, the functional requirement can be expressed in the following way:

F1. The IdPs and the SPs are part of the same federation.

Trust Requirements (TR): Jøsang et al. compiled a set of trust requirements for different Identity Management models [11]. Kylau et al. also present a set of requirements targeting mainly the FIM [13]. However, some trust requirements have been ignored by both works. We have picked the suitable existing requirements from [11, 13] and combined them with a few novel ones, that we have formulated, to create a comprehensive set of trust requirements for the FIM. The set is presented below. Each requirement has been marked to indicate if they existed before, as presented in [11, 13].

Client Trust in IdP:

T1. The IdP has implemented satisfactory user registration procedures and authentication mechanisms (from the client's perspective, denoted as *T2* in [11]).

T2. The IdP protects client privacy to the SP, either by using anonymous techniques or by using pseudonymous identifiers, when the client wishes to employ them (denoted as *T1* in [11]).

T3. The IdP has satisfactory mechanisms to store user attributes safely and securely.

T4. The IdP will release only those attributes to the SP that the client has consented to.

Client Trust in SP:

T5. The SP will ask only for the minimum number of user attributes that are required to access any of its services.

T6. The SP will not abuse the released user attributes and will use them only for the stated purpose(s).

IdP Trust in Client:

T7. The client handles their authentication credentials with adequate care (denoted as *T3* in [11]).

SP Trust in Client:

T8. During the registration procedure, the IdP might ask the client to provide several attributes which are then stored at the IdP. In such scenarios, the SP trusts the client to be honest while providing such attributes.

IdP Trust in SP:

T9. The SP adheres to the agreed privacy policies regarding non-disclosure of user data (denoted as *IdP-T.1* in [13]). In other words, the SP will not abuse the released user attributes and will use them only for the stated purpose(s). The policy might include that the SP will not cache any user-attributes other than those which are absolutely necessary. This is to ensure that the IdP can always provide the updated attributes regarding the user. In cases where the SP needs to cache any attributes (e.g. IdP-supplied identifiers), the SP must inform the IdP.

T10. The SP adheres to the agreed policies and procedures, in cases if they are available, regarding access control and delegated access. If there are no such policies or procedures, this (T10) requirement is ignored.

SP Trust in IdP:

T11. The IdP has implemented adequate procedures for registering users and for issuing credentials (denoted as *T7* in [11]).

T12. The IdP will authenticate the client appropriately as per the requirement and will release user attributes securely.

This set forms the basic set of trust requirements for the FIM. The attribute aggregation model leverages the basic FIM and extends it in many ways (which we will explore shortly). Such an extension requires formulating novel trust requirements which must be augmented with this basic set. We will formulate and analyse such additional requirements separately for each aggregation model.

Security Requirements (SR): Security requirements define what must be imposed to ensure the security of the system and all other parties involved. A taxonomy of requirements for an ideal Identity Management System can be found in [9]. We can use that taxonomy as a basis to formulate the list of security requirements which is given below:

S1. The user-registration procedure at the IdP is completed in a secure manner. If the registration takes place online, it must be done over the HTTPS channel.

S2. If the IdP allows the user to choose a credential (e.g. a password) for the identifier, the IdP must ensure that the chosen credential is secure (in other words, it is hard to guess or tamper with).

S3. The user is authenticated securely (over the HTTPS channel, if the authentication takes place online) using appropriate authentication mechanisms.

S4. The user attributes are stored securely at the IdP.

S5. To ensure the confidentiality, integrity and non-repudiation of the assertions/claims holding user attributes, they must be digitally signed and must be exchanged, between the IdP and the SP, over the HTTPS channel.

Privacy Requirements (PR): Privacy requirements mainly focus on preserving the privacy of the user while using an IMS. Using the taxonomy of requirements from [9], we can formulate the following list of privacy requirements:

P1. The IdP should provide the user with the choice to access services anonymously or using a pseudonym. If the user chooses to be anonymous, the user identifier should not be transferred to the SP. If the user chooses to use a pseudonym, an identifier (either a persistent or a temporary one) should be created and transferred to the SP.

P2. The IdP must inform the user about the SP to which the attributes are being released to ensure data transparency.

P3. The IdP must allow the user to select the attributes that they want to release to a particular SP. This will ensure that the user has consented to exchanging particular attributes and the user has full control over the data flow.

P4. To ensure data minimisation, the SP must inform the 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 a SP, is to allow users to administer their data remotely preferably using remote policies. If the IdP and the 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.

Analysis: The curious readers might have noticed many similarities between the Trust and Security-Privacy Requirements. In fact, functional, security and privacy requirements signify the technical conditions that must be considered while designing and developing an IMS and can be used to satisfy the trust requirements. Now we will analyse which Security and Privacy requirements can be used to satisfy which trust requirements.

The requirement *T1* can be satisfied by fulfilling requirements *S1, S2 & S3* since these security requirements ensure that the IdP has implemented the required methods for adequate user registration and subsequent authentication. It is easy to note that *P1* will fulfil *T2* and *S3* will fulfil *T3.* If *P2 & P3* can be fulfilled, this will end up satisfying *T4.* By ensuring *P4*, *T5* can be satisfied and similarly, by ensuring *P5*, *T6* can be fulfilled. If there are mechanisms available to allow remote administration of policies at the SP (*P5*), it will allow the IdP to ensure that the SP has adhered to the agreed policies regarding non-disclosure of user attributes and access control thereby fulfilling *T9 & T10.* Finally, it is not difficult to conclude that security requirements *S1 & S4* will help to satisfy *T11 & T12* respectively.

It is important to note that some requirements such as *T7 & T8* cannot be satisfied by any technical means since it is not possible to determine whether the user handles the credentials with adequate care or whether the user is honest in providing different attributes. In such cases, the IdP can only assume that the user will handle the credentials appropriately for the sake of their own interests or the SP can only trust that the user is honest while providing attributes - any of both such assumptions might not be honoured at all.

3. TAXONOMY OF ATTRIBUTE AGGREGATION MODELS

Attribute Aggregation is the mechanism of aggregating or collecting attributes of a user retrieved from multiple identity providers in a single session. In FIM technologies such as SAML, the attributes are embedded inside an assertion. Therefore we have two interpretations of attribute aggregation: it either means the aggregation of attributes when all attributes are embedded inside a single assertion or it means the aggregation of assertions when other assertions are embedded inside a single assertion.

Before we can create a taxonomy of existing attribute aggregation models, we need to select the criteria that can be used to classify each model. We have two criteria: where the attribute aggregation takes place and who mediates the whole process. Here, by *mediation* we mean the process of initiating the aggregation mechanism. Based on where the attribute aggregation takes place, the existing attribute aggregation mechanisms can be classified in three categories: Aggregation at the SP, Aggregation at the IdP and Aggregation at the Client (e.g. the User-Agent/Browser). Each category can then further be classified based on who mediates the aggregation process: the IdP or the SP. Based on these categories, the taxonomy that we have created is illustrated in Figure 1. In the following sections, we provide a brief description of each mechanism and analyse the functional, security and privacy requirements for each model.

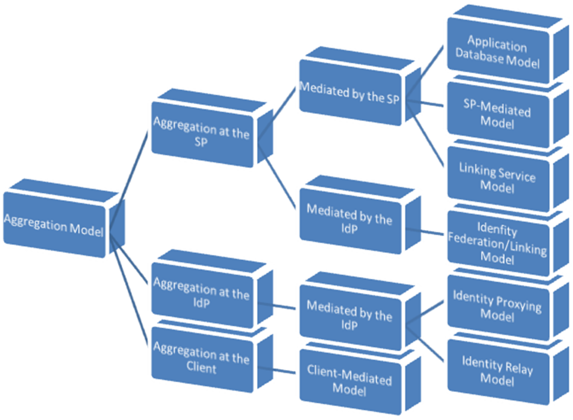

Figure 1: Taxonomy of Attribute Aggregation Models.

4. AGGREGATION AT THE SP

The models discussed in the section enable the user to aggregate attributes at the SPs. As mentioned earlier, the models can be further classified based on the entity that initiates the process: the SP or the IdP.

4.1 Mediated by the SP

The models which allow the SP to initiate the aggregation mechanism falls into this category.

4.1.1 Application Database (AD) Model.

This is the simplest form of attribute aggregation model (Figure 2) [12, 5]. In this model, the 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 the 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 on using this mapping to determine if the user is authorised to access a particular service.

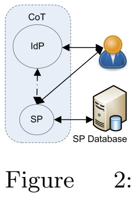

Figure 2: Application Database Model.

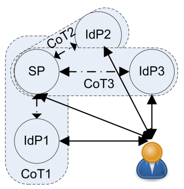

Figure 3: SP-Mediated Model.

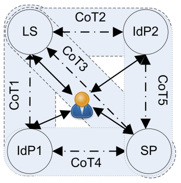

Figure 4: Linking Service Model.

FR: The functional requirement is that the IdP and SP are federated with each other meaning that the requirement *F1* must be fulfilled. In addition, the following requirement arises:

F2. The mapping of the IdP-supplied and the SP-created identifiers is correct.

TR: *T1 - T12*. The model formulates the following new trust requirement:

T13. The IdP provides a persistent SP-specific user identifier to the SP so that the SP can create the mapping between the IdP-supplied and SP-created identifiers in the first instance and then map the IdP-supplied identifier to the SP-created identifier in all subsequent instances to retrieve attributes from the local repository.

SR: *S1 - S5*. The model formulates the following new security requirement:

S6. The mapping information of the IdP-supplied and the SP-created identifiers as well as other local user attributes are stored safely and securely in the SP repository.

PR: The model does not introduce any additional privacy requirements. However, it is required that the existing privacy requirements (*P1 - P5*) are fulfilled.

Analysis - It is easy to note that the requirement *P1* satisfies *T13*.

4.1.2 SP-Mediated (SPM) Model.

In this model, the SP allows the user to aggregate attributes from multiple IdPs in a single session (Figure 3) [12, 5]. The user is forwarded to different IdPs one after another where the user is authenticated separately and returns 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.

FR: The functional requirement is that the SP and each IdP are federated with each other meaning that the requirement *F1* must be fulfilled for each IdP-SP pair. In addition, the following requirement arises:

F3. A session is maintained at the SP so that the SP can correlate the attributes from the current IdP with attributes retrieved previously from other IdPs.

TR/SR/PR: This model does not have any additional trust, security and privacy requirements other than *T1 - T12*, *S1 - S5* and *P1 - P5* respectively.

4.1.3 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, Figure 4) [7, 5] which is used by the 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 the SP, the user visits the SP and is forwarded to the first IdP (IdP1 in Figure 4). The user authenticates at IdP1 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 or 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.

FR: The functional requirement is that the IdPs and the SP, the LS and the IdPs and the LS and the SP are federated with each other meaning that the requirement *F1* must be fulfilled for each pair (except for any IdP-IdP pair). Moreover, *F3* has to be fulfilled during the Link Registration phase between the LS and the IdPs while keeping a session at the LS as well as during the attribute aggregation phase between the LS and other IdPs while keeping a session at the SP. In addition, the following requirements arise:

F4. The Linking Table is accurate so that the mapping of the IdP-supplied and the LS-supplied identifiers is correct.

F5. The LS has dual capabilities of an IdP as well of a SP. The LS has to act as an IdP to the SP and as a SP to other IdPs.

F6. The IdPs have extended capabilities to embed a referral to the LS and the encrypted persistent identifier for the LS along with the attributes inside an assertion to be sent to the SP.

F7. The SP has extended capability to follow the referral to interact with the LS so that the SP either can receive aggregated attributes from the LS or can receive a list of linked IdPs with their respective encrypted identifiers. For the second case, the SP must have a mechanism to retrieve attributes from the linked IdPs (except the first IdP) using the encrypted identifiers.

F8. The IdPs have extended capabilities to provide assertions containing attributes to the LS or the SP without any user authentication when the LS or the SP submits a valid encrypted identifier retrieved from the Linking Table of the LS.

TR: *T1 - T11.* In addition, the model has the following new trust requirements:

T14. The IdPs provide persistent LS-specific user identifiers to the LS so that the LS can build up the Linking Table and to the SP so that it can use it to query the LS.

T15. The LS builds up the Linking Table correctly to ensure that only IdPs to which a user has accounts are linked.

T16. The LS will either provide correctly aggregated attributes or the list of linked IdPs to the SP for that specific user.

T17. The SP will use that list to aggregate attributes just once.

T18. The IdPs will release attributes to the LS or the SP upon receiving the persistent identifier and without any user authentication.

SR: *S1 - S5.* In addition, the model has the following new security requirements:

S7. The Linking Table is stored safely and securely at the LS repository.

S8. Each LS-specific user identifier acts like a secret between the IdP and the LS. Hence, it must be encrypted by the IdP to be decrypted only by the LS and by the LS to be accessible by each individual IdP. This will ensure that the SP does not get hold of the respective identifier at any time.

S9. If the list of linked IdPs is released to the SP, the LS should ensure that the SP cannot use that list more than once.

PR: Other than fulfilling (*P1 - P5*), the model has the following privacy requirements:

P6. The IdP releases the referral of the LS to the SP only after the user has explicitly consented to do so.

P7. To ensure transparency, the SP should allow the user to choose where the attributes will be aggregated: at the LS or at the SP.

Analysis - Building an accurate Linking Table implicitly implies that the IdPs have provided LS-specific user identifiers to the LS. Also IdPs having the ability to embed the encrypted identifier means that the IdPs will release that identifiers to the SP. Therefore, *F4, F6 & P6* combinedly satisfy *T14.* It is easy to see that *F4* satisfies *T15*, *F7 & P7* satisfy *T16* and *S9* satisfies *T17.* Since all IdPs other than the first IdP where the user will authenticate herself will release user-attributes without any user-authentication, we will need *S8 & F8* to satisfy *T18.*

4.2 Mediated by the IdP

The model which allows the IdP to initiate the aggregation mechanism falls into this category.

4.2.1 Identity Federation/Linking (IFL) Model.

This model, introduced by the Liberty Alliance framework, is one of the very first models to address the problem of attribute aggregation (Figure 5) [12, 5]. In this model, IdPs allow the 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 a 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.

FR: The functional requirement is that IdPs and the SP are federated pair-wise meaning that the requirement *F1* must be fulfilled for each pair. Moreover, *F3* has to be fulfilled during the attribute aggregation phase between the SP and the other IdP while keeping a session at the SP. Moreover, *F6, F7 & F8* must be fulfilled using the random alias instead of the persistent identifier. In addition, the following requirement arises:

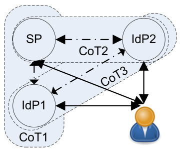

Figure 5: Identity Federation/Linking Model.

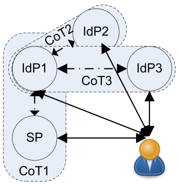

Figure 6: Identity Proxying Model.

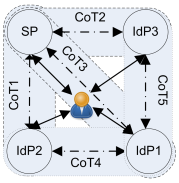

Figure 7: Identity Relay Model.

F9. The random alias is correctly created.

TR: *T1 - T11*. In addition, *T17* must be fulfilled using the random alias instead of the list and *T18* must be fulfilled using the random alias instead of the persistent identifier. The model also has the following new trust requirement:

T19. The IdPs provide the correct random alias to the SP.

SR: S1 - S5. In addition, the model must also satisfy *S8* where the random alias needs to be encrypted instead of the persistent identifier. Moreover, *S9* has to be met to ensure that the SP cannot use the random alias to aggregate attributes more than once in a single session. The model also has the following new security requirement:

S10. The random alias is stored safely in the IdP repository and is transmitted securely during any transmission.

Privacy Requirements - The model must fulfil requirements *P1 - P6*.

Analysis - It is easy to see that *F9* satisfies *T19*.

5. AGGREGATION AT THE IDP

The models discussed in the section enable the user to aggregate attributes at the IdPs.

5.1 Mediated by the IdP

The model that allows the IdP to initiate the aggregation mechanism falls into this category.

5.1.1 *Identity Proxying (IP) Model.*

In this model, the SP allows the user to aggregate attributes from multiple IdPs using a highly trusted IdP (Figure 6) [12, 5]. The user is forwarded to the trusted IdP (IdP1 in Figure 6) 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.

FR: The trusted IdP is federated with other IdPs and the SP meaning that it needs that the requirement *F1* is fulfilled for the trusted IdP with other IdPs and the SP. Moreover, *F3* has to be fulfilled during the attribute aggregation phase between the trusted IdP and other IdPs while keeping a session at the trusted IdP. Also, *F5* has to be fulfilled in the sense that the trusted IdP should have the dual capabilities: acting as an IdP to the SP and acting as a SP to other IdPs. In addition, the following requirement arises:

F10. The assertion returned by other IdPs should be targeted for the trusted IdP so that it can validate each assertion, extracts attributes from them and then aggregate all of them, possibly also with its own attributes.

TR/SR/PR: This model does not have any additional trust, security and privacy requirements other than *T1 - T12*, *S1 - S5* and *P1 - P5* respectively.

5.1.2 *Identity Relay (IR) Model.*

The Identity Relay model is a generalised case of the Proxying model (Figure 7) [12, 5]. Since the Proxying model requires the 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 (or Relay IdP), (IdP1 in Figure 7) is used instead of a trusted IdP . The 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 assertions including user-attributes. The relay IdP combines all assertions into a single assertion and forwards it to the SP. The SP extracts embedded assertions from this assertion and validates each assertion to retrieve attributes from other IdPs. Based on the combined attributes, the SP determines if the user can access the service.

FR: The functional requirement is that the IdPs and the SP, the relay IdP and other IdPs and the relay IdP and the SP are federated with each other meaning that the requirement *F1* must be fulfilled for each pair (except for any IdP-IdP pair). Moreover, *F3* has to be fulfilled during the attribute aggregation phase between the relay IdP and other IdPs while keeping a session at the relay IdP. Also, *F5* has to be fulfilled in the sense that the relay IdP should have the dual capabilities: acting as an IdP to the SP and acting as a SP to other IdPs. In addition, the following requirement arises:

F11. The assertion returned by other IdPs should be targeted for the SP. The relay 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.

TR: Other than fulfilling (*T1 - T12*), the model has the following trust requirement:

T20. The other IdPs release assertions in such a way that they are only accessible by the SP.

SR/PR: This model does not have any additional security and privacy requirements other than *S1 - S5* and *P1 - P5* respectively.

Analysis - It is easy to see that *F11* satisfies *T20*.

6. AGGREGATION AT THE CLIENT

In this model, a user client (e.g. a browser) is used to aggregate attributes.

6.0.3 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 [12, 5]. The 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.

FR: The functional requirement is that the IdPs and the SP are federated with each other meaning that the requirement *F1* must be fulfilled for each pair (except for any IdP-IdP pair). Also, *F11* must be fulfilled where all intermediary functions should be performed by the intelligent client rather than the relay IdP. In addition, the following requirement arises:

F12. The intelligent client, the IdP and the SP have been deployed with the required capabilities to interact with each other.

TR: This model requires *T1 - T12* and *T20* to be fulfilled.

SR/PR: This model does not have any additional security and privacy requirements other than *S1 - S5* and *P1 - P5* respectively.

7. DISCUSSION

We have shown in the previous section that the aggregation models formulate novel trust requirements. They also require novel functional, security and privacy requirements. The previous section has analysed how these novel functional, security and privacy requirements can satisfy the newly formulated trust requirements.

Each model has different sets of requirements with their own strengths and weaknesses. It is rather difficult to illustrate this fact with any textual description as provided in the previous section. Therefore, we present our analysis in tabular forms below. Table 1 and Table 2 illustrate a side-by-side comparison of all aggregation models where Table 1 presents the comparison of requirements and Table 2 presents the comparison of strengths and weaknesses of each model. The more the 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. Unfortunately, it is evident from the tables that no model is unquestionably superior to other models. Extensive research is needed to come up with a model that requires not only a relatively small number of requirements but also offers a good number of advantages. In our opinion, a good research candidate for that would either be the Identity Proxying model or the Identity Relay model. We are currently investigating how these two models can be implemented while reducing their disadvantages as much as possible.

8. RELATED WORK

The works discussing different existing attribute aggregation mechanisms can be found in [12, 5] where the author provided the comparative analysis of existing attribute aggregation models and an exhaustive discussion on several aspects of attribute aggregation and identity management in general without using any requirement. In another work, the authors presented a survey of requirements required for attribute aggregation from multiple sources [10] and analysed the strengths and weaknesses of four different generic mechanisms (Identity Proxying, SP-Mediated, Client-Mediated and Identity Federation/Linking) using that set of requirements.

Our work in this paper is influenced by the works presented in [12, 5, 10]. However, we believe that our work is more comprehensive in every sense. The set of requirements devised in this work is more thorough than that of [10] where the trust, functional, security and privacy requirements were not properly categorised and many trust, functional, security and privacy requirements were not even considered. The works of [12, 5] just provided a brief discussion on each model without any analysis of requirements. Our work is the first one that provides a taxonomy of existing attribute aggregation models, categorises different requirements in a systematic manner, analyses the novel requirements that arise for each aggregation model and interlinks how different trust requirements can be fulfilled using the functional, security and privacy requirements. It also presents the result in tabular forms to compare all models side-by-side not only based on their architectural flexibilities/constrains, but also based on different requirements.

9. CONCLUSION

As online services evolve and their maturities increase, there will be a pressing need for a secure yet usable attribute aggregation mechanism. As our findings suggest, none of the existing methods can claim to be unquestionably better than others and each has major drawbacks that must be addressed. In essence, there is a need for further research on it. For that, a thorough analysis of existing methods are necessary to make readers to be aware of the additional requirements and to keep them informed of the strengths and weaknesses of each model, thereby enabling them to gain a deep understanding on each one. In this paper, we aim to address these issues. We have created a taxonomy of existing attribute aggregation models and analysed them individually to investigate the additional requirements each model has. Then we have presented our finding in tabular forms to make it easier for the reader to compare each model side-by-side. We believe that this work will lay down the path to help anyone to embark on research to design and develop a secure, usable and realistic attribute aggregation mechanism.

10. REFERENCES

[1] Shibboleth. `http://shibboleth.internet2.edu/`.

[2] SimpleSAMLphp. `http://simplesamlphp.org/`.

[3] OpenID Authentication 2.0 - Final. 5 December, 2007. `http://openid.net/specs/openid-authentication-2_0.html`.

[4] Modinis - Common Terminological Framework for Interoperable Electronic Identity Management. Accessed on 28th June, 2011. `https://www.cosic.esat.kuleuven.be/modinis-idm/twiki/bin/view.cgi/Main/GlossaryDoc`.

Table 1: Aggregation Models: Comparison of Requirements

Models	TR	FR	SR	PR
AD	T1 - T13	F1 - F2	S1 - S6	P1 - P5
SPM	T1 - T12	F1, F3	S1 - S5	P1 - P5
LS	T1 - T11, T14 - T18	F1, F4 - F8	S1 - S5, S7 - S9	P1 - P7
IFL	T1 - T11, T17 - T19	F1, F3, F6 - F9	S1 - S5, S8 - S10	P1 - P6
IP	T1 - T12	F1, F3, F5, F10	S1- S5	P1 - P5
IR	T1 - T12, T20	F1, F3, F5, F11	S1 - S5	P1 - P5
CM	T1 - T12, T20	F1, F11 - F12	S1 - S5	P1 - P5

Table 2: 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 based on Shibboleth and SimpleSAMLphp.
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 based on Shibboleth and SimpleSAMLphp.
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.

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

[6] David W Chadwick. Federated Identity Management. In A. Aldini, G. Barthe, and R. Gorrieri, editors, *FOSAD 2008/2009*, number 5705 in LNCS, page 96-120. Springer-Verlag, Berlin, January 2009.

[7] D.W. Chadwick and G. Inman. Attribute aggregation in federated identity management. *Computer*, 42(5):33-40, 2009.

[8] David Chappell. Introducing Windows CardSpace, April 2006. http://msdn.microsoft.com/en-us/library/aa480189.aspx.

[9] M.S. Ferdous and R. Poet. A comparative analysis of Identity Management Systems. In *High Performance Computing and Simulation (HPCS), 2012 International Conference on*, page 454-461, july 2012.

[10] George Inman, David W Chadwick, and Nate Klingenstein. Authorisation using attribute from multiple authorities-a study of requirements. In *Proceedings of HCSIT Summit-ePortfolio International Conference*, 2007.

[11] Audun Jøsang, John Fabre, Brian Hay, James Dalziel, and Simon Pope. Trust requirements in identity management. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research - Volume 44*, ACSW Frontiers '05, page 99-108, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[12] N. Klingenstein. Attribute Aggregation and Federated Identity. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, page 26-26, 2007.

[13] U. Kylau, I. Thomas, M. Menzel, and C. Meinel. Trust Requirements in Identity Federation Topologies. In *Advanced Information Networking and Applications, 2009. AINA '09. International Conference on*, page 137-145, 2009.

[14] Md. Sadek Ferdous, Mohammad Jabed Morshed Chowdhury, Md. Moniruzzaman, and Farida Chowdhury. Identity federations: A new perspective for Bangladesh. In *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, page 219-224, may 2012.

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