# Authorisation using Attributes from Multiple Authorities – A Study of Requirements

David Chadwick[1], George Inman[1], Nate Klingenstein[2]
[1]Computing Laboratory, University of Kent, UK. [2]Internet2 Consortium, USA

## Abstract

This paper presents the results of a survey of requirements for attribute aggregation in authorisation systems, gathered from an international community of security professionals. It then analyses these requirements against 4 generic models for attribute aggregation and makes some recommendations for future implementations.

## 1. Introduction

Due to the global nature of today's networked resources the need for flexible and easily managed authorisation infrastructures is increasingly important. Many grid and campus based network applications are now being enabled with attribute based authorisation (ABAC) [1], in which users are granted access to network resources based on their various attributes e.g. their university affiliation, role within a virtual organisation (VO) or society membership. ABAC means that any user with a set of valid attributes will be granted access to a particular service. Although the model allows attributes to be retrieved from anywhere, in practice the set of attributes will usually be provided by a single entity commonly known as the Identity Provider (IdP) or Attribute Authority (AA). This IdP or AA will typically have access to a local database containing attributes about known users, along with the local name or identity of each user. Whilst this technology is increasingly being implemented in new grid and campus based applications to allow distributed access to network based resources from anywhere in the world, there are limitations with the current implementations, in that most ABAC systems receive all the user attributes from a single IdP, which limits the technology to receiving just one set of user attributes. This is one of the limiting factors of Microsoft's Cardspace [2]. Researchers and early adopters are realising that a single source of user attributes is insufficient for authorisation in many applications e.g. access to a medical database might require a GP attribute from the General Medical Council and a consultant attribute from the employing hospital; or online shopping might require a credit card from a bank for the purchase and a frequent flyer card to award air miles. Since the same user will usually be known by different identities in each IdP/AA, this makes the collection and aggregation of attributes from different IdPs/AAs difficult.

Before embarking on developing a solution to the attribute aggregation problem, we first thought it would be beneficial to determine a set of system requirements. We realised that determining end user (i.e. consumer or customer) requirements directly from the end users would be difficult at this stage, since most end users would not be sufficiently conversant with the problem space or terminology to present us with their considered requirements. Consequently the people we surveyed were security professionals who are already working in the general area of network authorisation and virtual organisations, and who are already aware of this problem space. We designed a questionnaire to explore the requirements for a new ABAC system that can be used to query multiple attribute authorities and return a set of aggregated attributes based on the multiple sets returned from the AAs. The methodology we employed, and the results of the questionnaire responses are presented in Section 2. In Section 3 we describe 4 generic models for attribute aggregation and analyse these requirements against those models. Section 4 concludes with our future plans for work in this area.

## 2. Determining User Requirements

### *2.1 Methodology*

There are various ways of determining requirements e.g. structured interviews (face to face or via the telephone), focus groups and questionnaires. Since our respondents are distributed around the globe, face to face interviews and focus groups were not feasible. Even telephone interviews would be difficult given the large time zone differences between the participants. Consequently we decided that a questionnaire distributed by email was the most appropriate tool for eliciting requirements.

There were several different sets of requirements that we wished go gather, for example, privacy requirements, trust requirements and protocol requirements. Consequently the questionnaire was divided into six sections. The first section attempts to capture general requirements in terms of the perceived need for attribute aggregation in both the short to medium term, the likely number of IdPs that will need to be aggregated and the typical end users of attribute aggregation. The second section determines the privacy requirements that any attribute aggregation authorisation system will need to meet. The third section determines the control requirements for attribute aggregation, in terms of who should have the power to decide which attributes can be aggregated either in a user session or independent of any session. The fourth section determines the protocol requirements for collecting the attributes. The fifth section determines the trust requirements between the various entities involved in attribute aggregation, and the need for attribute (credential) signing and dynamic delegation of authority. The last section was a catch-all that allowed the users to supply any additional requirements they might have that had not been covered in the previous sections of the questionnaire. It also allowed the respondents to provide their use cases for attribute aggregation and optionally their demographic information. The questionnaire comprised 23 questions in all.

When questions needed to elicit a respondent’s opinion about a topic, we usually used a Likert-type 5-point scale, with answers ranging from i) Of no importance at all, ii) Probably not that important, iii) Potentially important (50/50) iv) Important v) Very Important/Essential. Sometimes we added a sixth option of Don’t Care when this seemed like a sensible choice. Other questions required users to choose one or more of several options e.g. whether digitally signed assertions should be available in all protocol exchanges, or only in some or in none at all.

The draft questionnaire was distributed to six people who had a close relationship with the project team, in order to test its semantic clarity, lack of ambiguity, effectiveness and coverage of the topic. Half of these respondents provided useful feedback to improve several of the questions.

The survey was distributed to members of 12 international mailing lists (see Appendix 1). We received 26 replies within the allotted timeframe, and a summary of the results is presented below. Unfortunately we cannot provide an accurate response rate since we do not know the sizes of the various mailing lists, but it is likely to be less than 10%.

75% of the respondents said they had an above average or very good knowledge of computer security compared to the average computing professional. Only 8% (2 respondents) said they had below average or very little computer security knowledge. Over 50% of the respondents had more than 10 years of experience as a computing professional, with 13% having more than 25 years of experience. The vast majority of the respondents (92%) were from education or research sectors with just 2 respondents being from the commercial sector.

The full questionnaire can be obtained from http://sec.cs.kent.ac.uk/shintau/Questionnaire.doc

## 2.2 General Requirements

The first section was designed to gauge the importance of authorisation based on attributes from multiple attribute authorities, both now in the medium term (2-5 years), as well as how many attribute sources are likely to be needed by future authorisation systems. We also wanted to determine who the likely users of such systems would be.

65% of respondents felt that it was either important or very important at the present time, compared to just 7.6% who felt that it was probably not that important or of no importance at all. In the medium term the importance increased, with 91.7% of participants believing that it will become important or very important to them. There was also a doubling in those who felt that it would become very important to them, rising from 30.8% to 62.5%.

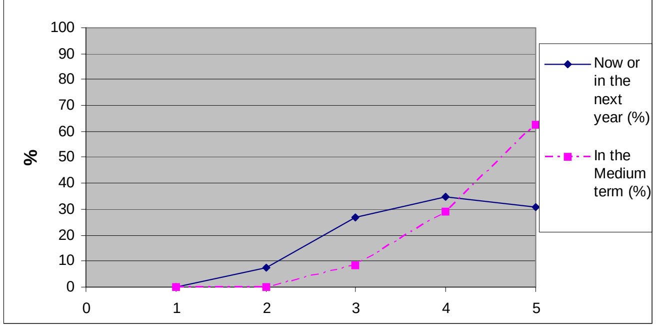


Figure 1. Importance of authorisation using attributes from multiple sources

We then asked how many attribute sources would be likely to be required in any one user's authentication session. Participants were given 4 options to choose from on a sliding scale from one to more than three. The results showed that 54% of participants thought that more than 3 attributes sources could be required compared to only 4% that believed a single attribute source would be sufficient.

Finally we asked who are likely to be the typical users of any attribute-based authorisation infrastructure. We provided three options: Humans via Web Browsers, Applications via APIs and Grid users via Grid clients. Participants were allowed to choose multiple values and could also specify their own users for the system. The results showed a broad spread with all of the provided options being potential users. 80% of participants believed that Humans via web-browsers would be potential users, 77% believed that grid users and their clients would be potential users and 65% believed that Application and API's would be potential users. Other suggested users were: Smart network devices to enable inter-operation with users and devices, Intermediaries such as online CAs, grid portals/gateways, IdP proxies, Experimental data collection systems, Shibboleth type enabled SPS, Command Line Interfaces and in silico computing. Consequently any provided solution should have a broad range of applicability, and cannot assume any single model of use, such as a web browser will always be involved.

## 2.3 Privacy Requirements

The second section dealt with the user's privacy within an attribute based environment. We began by questioning the importance of privacy protection for user attributes using the five point Likert scale from "of no importance" to "very important/essential". The results showed that 62% of participants believed this issue to be important or very important compared to 14% which believe it be of little to no importance.

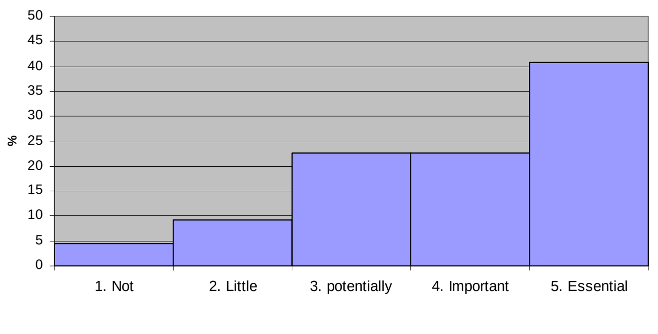


Figure 2. Importance of Privacy Protection

We asked how the privacy of user attributes should be enforced. Participants were given a choice of 4 options (and their percentage responses are shown in parentheses):

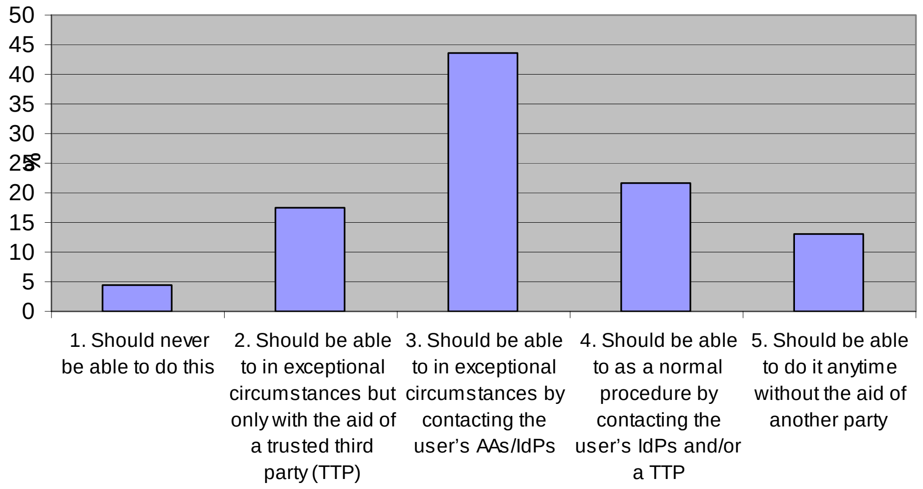
1. Legal enforcement is sufficient. No technical controls are needed (4%)
2. Legal enforcement should be supplemented with some technical controls (26%)
3. Technical controls should be used to enforce all legal requirements (18%)
4. Technical controls are essential and should be independent of legal matters (52%)

All but 4% of the respondents stated that some technical controls should be implemented and the majority believe that the controls should be independent of legal matters.

We asked whether service providers should be able to track users between sessions even if they did not know the true identity of the users (e.g. if pseudonyms were being used). Participants were asked to choose one of five Likert scale options ranging from; should never be able to do this to very important/essential for to do this for all applications. The results were that none of the respondents felt that the SP should never be able track users between sessions and only 14% thought that this was essential for all applications. The majority of participants (56%) believed that a service provider (SP) should be able to track a user between sessions for most applications (but not for all), meaning that this must be an option of any attribute aggregation protocol.

We then asked whether a service provider should be able to learn the true identity of users, the vast majority (96%) believing that the SP should be able to, but the "when and how" differed (see Figure 3). 13% felt that the SP should be able to access a user's identity anytime without the aid of another party, whilst the majority (43%) thought the SP should only be able to do this in exceptional circumstances by contacting the user's AAs or IdPs. Clearly the "when and how" has an effect on the design of the attribute aggregation protocols.

**Figure 3. How important is it that a service provider can learn the true identity of a user**



We asked whether AAs/IdPs should be able to communicate with each other in order to link together the attributes of a user. Participants were asked to choose one of the following (and their percentage responses are shown in parentheses):

1. Yes, and without the aid or permission of the user (19%)
2. Yes, but only with the permission of the user (62%)
3. Yes, but only with the technical aid of the user (15%)
4. No, it should not be technically possible (4%)

The results clearly show that the majority (77%) believe that AAs/IdPs should only be able to communicate with each other with the permission or aid of the user.

The final question in this section asked whether SPs should be able to search or query multiple AAs/IdPs in order to look for linkages between user attributes. Participants were asked to choose one answer from the following (and their percentage responses are shown in parentheses):

1. Yes, anytime it wants to (0%)
2. Yes, but only if it needs to pull more attributes in order to authorise the user (28%)
3. Yes, but only if it needs to pull more attributes in order to authorise the user, and then only with the user's permission (60%)
4. No it should not be technically possible (12%)

Clearly any “pull” protocol design should be cognisant of the fact that a user must have given their permission first before an SP is allowed to pull additional attributes.

## 2.4 Control Requirements

This section was designed to establish who should be in control of attribute aggregation and the definition of the various attributes that are needed for authorisation.

The first question looked at whether there should be a master list of all the IdPs of a given user and all the attributes that they hold, and if so, who should control this master list. Participants were asked to choose between 6 values (percentage responses in parentheses); the user only (31%), an agent trusted by the user (15%), the user’s primary IdP (19%), distributed between multiple IdPs (19%), each service provider (0%) and a third party directory service (15%). The results were fairly evenly spread, except that everyone agreed that the SPs should not hold such a list, and a slight preference was given to only the user knowing who all his IdPs are.

The next question looked at which party should be responsible for controlling the aggregation of a user’s attributes in an authorisation session. Participants were given the following options and were allowed to choose multiple values (their percentage responses are shown in parentheses):

1. the user should collect together the necessary attributes and push them to the service provider (42%)
2. the user should collect together references to the appropriate attributes and push these to the service provider for it to pull the attributes (33%)
3. the user should contact an intermediate gateway that will collect (pull) the attributes on his behalf and push them to the service provider (33%)
4. the user should simply contact the service provider and the infrastructure will know which attributes to pull from where (42%)
5. other mechanism (8%)

Clearly there is no preference for either “push” or “pull” modes of attribute collection, or whether users, SPs or intermediaries should do the aggregation. Two of those questioned offered additional mechanisms for this process: “The user collects together the necessary attributes and pushes them to the service provider through a trusted agent” and “for Institutions and IdP maintainers to provide well thought out policies and mechanisms for genuine informed consent”.

The final question attempted to find the correct balance of power between SPs and IdPs over the sets of attributes that are needed for application authorisation. Participants were asked to choose one of the following 5 options (and their percentage responses are shown in parentheses):

1. The SP should publish policies about what attributes it needs and the IdPs/AAs should be capable of issuing these attributes (22%)
2. The IdPs/AAs should publish policies about what they can issue, and the SPs should build systems that make use of them (15%)
3. There should be prior negotiation between the SP and the IdPs/AAs and they should mutually agree which attributes are needed for each application. (22%)
4. There should be an internationally standardised set of attributes used by all IdPs/AAs and SPs (26%)
5. Other (15%)

As can be seen, no option shows significant preference, and the power to control is fairly evenly distributed between both IdPs and SPs.

## 2.5 Protocol Requirements

This section ascertained what types of protocol should be implemented by any attribute aggregation system. The first question asked whether tunnelling through firewalls (using http or https) was important. Everyone had an opinion about this. 79% said this was either essential or should be done if possible.  17% said this wasn't really necessary and only 4% thought it was very undesirable. We conclude that there is a very strong bias for a http based protocol.

The next question asked whether the pull protocols should be based on web services/XML/SOAP. Participants were given the same five choices as in the question above. 67% said this was either essential or should be done if possible. 8% said it wasn't really necessary, no-one thought it was undesirable, and 25% didn't care. We conclude that an XML/SOAP based message is the favoured approach.

We then asked whether existing protocols should be used and if so whether they should be extended in a standard way for interoperability. The results showed clearly that the vast majority of participants (88%) would prefer the use of existing protocols (44% said it was essential and 44% said Yes if possible). Only 8% didn't care and 4% thought it unnecessary. We also asked whether it would be excusable to break or extend (in a non standard way) existing protocols in order to achieve the required functionality. 33% said this was very undesirable, and 58% cautioned "only if really necessary". Only 8% felt that standard protocols could be broken to achieve the requirements. Clearly standards conformance is a very important issue.

Participants were asked to suggest relevant protocols to be used. Thirteen different protocols were suggested, but the most commonly nominated protocol was SAML with 31% of the votes.

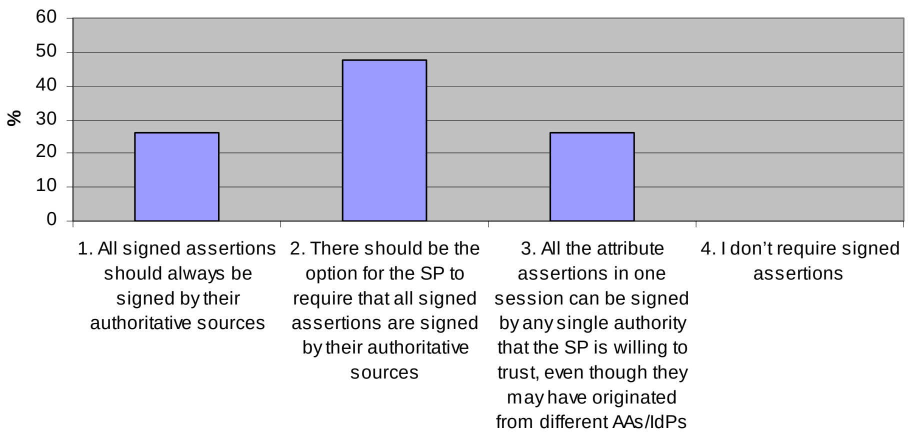
Finally the respondents were asked if some form of proxying of identity information should be supported. The majority of users (70%) felt that this was an essential feature, with 20% thinking that a single proxy would be sufficient but 50% believing that multiple proxys (or protocol hops) must be supported. A further 15% said proxying should be supported if possible. Only 5% thought proxying was undesirable and 10% didn't care.  Clearly any attribute aggregation system will need to support proxying if it is to be widely accepted.

### 2.6 Trust Requirements

This section dealt with the trust issues surrounding the use of multiple IdPs. The first question asked whether it was important that each IdP be able to sign the assertions that it issues in order to allow a relying party to prove their validity. Participants were given the choice of 3 options (and their percentage responses are shown in parentheses):

1. signed assertions never need to be supported (0%)
2. the ability to sign assertions is needed for some messages (25%)
3. the ability to sign assertions needs to be supported for all exchanges (75%)

**Figure 4. Should the SP be able to validate the authoritative sources**

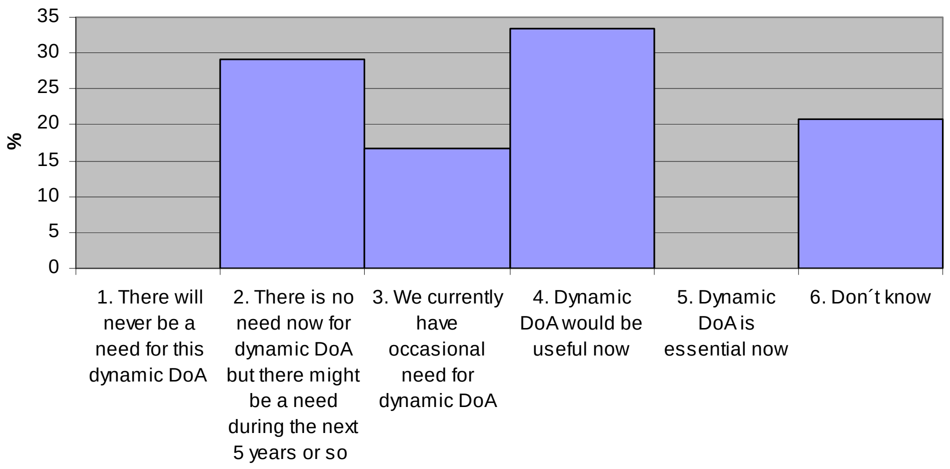


Signed assertions are clearly an essential component of attribute aggregation.

The next question attempted to ascertain whether service providers should be able to validate that the user’s attribute assertions were actually signed by their various authoritative IdP sources. The results are shown in Figure 4. The results show that all participants believe that assertions should be signed in some manner, and that 47% believe that the SP should be able to require the assertions be signed by their authoritative sources.

The final question asked if dynamic delegation of authority (DoA) was important. By this we mean that an authoritative source for an attribute can dynamically delegate to subordinates to sign the assertions on their behalf without informing the relying parties first. E.g. in a VO, the VO manager can decide to let various site managers issue VO membership certificates on his behalf, whilst the relying parties (SPs) simply say they trust the VO manager (and his delegates) to issue VO membership certificates. Participants were given the choice of 6 options and there responses are shown in Figure 5 below.

**Figure 5. Support for dynamic delegation of authority**



The results show that 33% of those questioned felt that dynamic DoA would be a useful feature to have now and that 17% currently have the occasional need for it. It is a feature that is likely to be increasingly needed in the future.

## 2.7 Additional Requirements and case studies

In this section we asked participants to provide us with any additional requirements and use cases they may have.

We received 13 additional requirements or requests for further discussion of parameters. These requirements were for any authorisation system to be interoperable with HEI in the US, EU and the wider world, for explicit testing of “novel” and unusual situations, for support for multiple sources of data that require authentication due to licensing restrictions, for a method to allow users to see who is using their data and for what purposes, for the designed systems to be simple enough to be usable, to allow IdPs to attach limitations on usage on assertions given to a SP and to provide a mechanism within delegation to know not only the target’s identity but also the issuer’s identity. The requests for further consideration of requirements pertained to the relationships between parties that wish to do collaborative research under different authentication regimes, the form of attribute aggregation, a need for further consideration of pushed attributes and a request to look into the requirements of differing communities in order to ultimately produce a system that can be used by them all.

We also asked participants to submit their current or future use cases in order to be able to best judge the way in which they would use any authorisation system produced. The common themes of the submitted use cases were the use of grid computing in projects as well as Shibboleth and federated access to resources. Virtual organisations were also mentioned often as well as access to confidential information such as NHS or government records.

### 2.8 Summary of Requirements

In summary, the following requirements are seen to be important by the majority of the respondents for any new multi-source attribute authorisation system:

1. Attribute aggregation must be usable in a variety of ways: Humans via web browsers, Applications via APIs and Grid users via grid clients etc.
2. Privacy protection of user attributes is of high importance and this should be through the use of technical controls, which are independent of legal means.
3. Service Providers should be able to track users between sessions if required
4. Service Providers should be able to learn the true identity of users in exceptional circumstances, but only by contacting the user's IdPs.
5. AAs/IdPs should only be able to communicate with each other to link together the attributes of a user with the user's permission.
6. Service providers should only be able to query multiple identity providers, in order to pull additional attributes for authorisation purposes, with the user's permission
7. Should be able to tunnel through firewalls using existing open ports (http/https)
8. System should use existing standard protocols and only extend them in a standard way if necessary. SAML is the most popular choice.
9. The proxying of information should be supported through multiple hops/proxies
10. The ability to sign assertions should be supported for all exchanges
11. The SP should be able to require that all assertions are signed by their authoritative sources
12. Should be easy to use by end-users and have the minimum amount of interaction[1]

Unfortunately some of these requirements are mutually exclusive i.e. 9 and 6/11. In general it is not possible to support multi-hop proxying, where entities on one side of the proxy are not always aware of the entities on the other side of a proxy, and to have attribute assertions that are always signed by their authoritative sources and to have the SP directly query multiple IdPs.

## 3. Analysis of Requirements against Attribute Aggregation Models

In this section we compare the 12 requirements derived above with those offered by 4 generic models which we have distilled from the literature. We believe that most, if not all, models of attribute aggregation are variations on these 4 generic models.

### 3.1 IdP Chaining

In the IdP chaining model multiple IdPs are accessed in succession before a single set of assertions is returned to the SP. This is the model as typified by myVOCS [4]. Each intermediary IdP in the chain is a combination of both an IdP and a SP as it both receives and issues attribute assertions. The initiating SP redirects the user to the first intermediary IdP, which redirects the user to the next intermediary IdP and so on down the chain until the terminating IdP is reached. The user is then authenticated by the terminating IdP, and is redirected back up the chain to the SP co-located with the last intermediary IdP in the chain. This redirection response contains an authentication assertion from the terminating IdP and may contain attribute assertions as well. The SP at this intermediary IdP redirects the user to the IdP component co-located with it, asking the user to authenticate to this IdP. The IdP notes that the user has already been authenticated by the terminating IdP, and therefore issues its own authentication assertion along with its own attribute

[1] This last requirement was not part of the questionnaire, but was mentioned by at least one respondent, and should be a "given" for any system that is to gain wide acceptability

assertion which will include any attributes provided by the terminating IdP. The user is then redirected back up the chain to the SP co-located with the next intermediary IdP in the chain. Eventually the user is redirected back to the initiating SP, by which time his request contains an authentication assertion and attribute assertion issued by the first intermediary IdP in the chain. The attribute assertion potentially contains attributes from each IdP in the chain.

This model can be seen to have a low level of protection for user attributes as every intermediary IdP must relay bearer credentials intended for a third party allowing for an increased risk of substitution attacks as well as the possibility of an IdP sending false authentication information through the chain causing the wrong users attributes to be released to the SP. As this model uses trust relationships between linked IdPs rather than explicit trust links with the SP all relevant attribute assertions will be returned to the SP by the last IdP in the chain regardless of what circles of trust the other IdPs in the chain might belong to. There is therefore an implicit trust relationship between every IdP in the chain and the SP, even though the SP may not be aware of it. This model allows the SP to track users between sessions if the same first intermediary IdP is used in each request. In this model we assume that each IdP-IdP link is initialised only with the user's permission, but it may not be obvious to the user what chains exist between IdPs. This model is the only one that allows for the use of multi-hop proxying as each link in the chain can be seen to be a proxy hop. Assertions signed by their authoritative sources could be supported but the protocol becomes more complex. This model requires the use of browser interactions and requires a medium to high level of user interaction.

### *3.2 SP-Mediated Attribute Aggregation*

This model is an enhancement of the Shibboleth model [5], in which the SP now queries multiple IdPs, rather than just one, in order to obtain their attribute assertions. As each IdP is contacted, the user is invited to authenticate to it. This model can be seen to offer excellent levels of privacy protection as the user must authenticate at each IdP, fully controls the attribute linking, and each set of assertions can be encrypted for the SP. The assertions are also signed by their respective authoritative sources. As this model is primarily SP based the requirement for SPs to be able to track users between sessions is easily accomplished as is the requirement for SPs to be able to learn the true identity of users. This model however precludes the use of multi-hop proxying as attributes are explicitly requested using a SP-IdP trust relationship. Unfortunately this model requires browser based technologies and requires a high level of user interaction as users must authenticate themselves at each IdP via redirects from the SP before the attributes are returned.

### *3.3 Client Based Collection*

The Client-Based assertion collection model is an enhancement of the model utilised by Microsoft's Cardspace [2] so that multiple IdPs are contacted instead of just one. When each IdP is queried, the user authenticates to it and a set of attributes are returned. This model requires a smart client that is able to create the attribute requests and collect the returned assertions into a single bundle to forward to the SP. The assertions obtained by the UA may be encrypted for the SP only, so that the UA or any intermediate nodes cannot read them. There is a high level of privacy protection for the user attributes, the assertions are signed by their authoritative sources, and the user is in control of the attribute aggregation. Unfortunately multi-hop proxying is precluded by this model as explicit requests must be made from the UA to each IdP, which then issues attribute assertions for the SP preventing the use of IdPs that are unknown to the UA or SP. This model also mandates the use of a smart browser or smart client that is able to make the attribute requests and store the returned assertions until the complete set of assertions have been obtained.

### *3.4 Identity Linking*

The identity linking model relies on the ability of a user to associate identities that it controls, prior to invoking any SP. If a user authenticates successfully as different identities to two different IdPs, it can claim control of both identities and request that one identity be federated with the other. When this is done, a unidirectional persistent identifier is created that allows one IdP to point to the counterpart identity at the second IdP. This may be repeated with the IdPs swapped to create a bidirectional link with two distinct identifiers [3, 6]. When a user contacts an SP for a service, it is redirected to one of the IdPs for authentication, and this provides the SP with the user's attributes that it holds plus the unidirectional link to the second IdP so that the SP can retrieve additional attributes from there. A variation on this model is to have an IdP *discovery service* that holds links to all the user's IdPs [7], rather than having multiple IdP-IdP links.

This model offers high levels of privacy protection for user attributes by ensuring that all attribute assertions are signed by their authoritative sources, and IdP attribute sets are only linked with the users permission. However it requires the SP to have more trust in the IdPs when they hold links to other IdPs, and to have a high level of trust in the central discovery service as it contains links to all the user's identification details for each IdP as well as potentially a list of attributes stored at each IdP. This model also implicitly requires that each IdP trusts every other linked IdP to authenticate principals correctly. This model allows SPs to track users between sessions to find out their true identities. Multi-hop proxying is unsupported as the initial IdP encrypts the request to the linked IdP or discovery service using its public key, preventing it from being passed to any other service. As each attribute assertion is encrypted to the SP there must be explicit trust links between the SP and each IdP that issues the attribute assertions. The level of user interaction can however be seen as quite low as users are only required to authenticate at a single IdP and that authentication request is used by the system to issue attribute requests to other IdPs. Due to this low level of user interaction there is no need for this model to require the use of a browser based client.

### 3.5 Requirements Analysis

The table below summarises how each of the 4 models satisfies the 12 requirements presented in section 2.8. 1 indicates the requirement is satisfied, 0 that it is not satisfied

**Table 1 – Requirements Matrix**

| **Requirements** | **IdP Chaining** | **SP Mediated Aggregation** | **Client Based Collection** | **Identity Linking** |
|---|---|---|---|---|
| 1. Does not mandate client interaction | 0 | 0 | 0 | 1 |
| 2. Privacy Protection of user attributes | 0 | 1 | 1 | 1 |
| 3. Service Provider able to track a user between sessions | 1 | 1 | 1 | 1 |
| 4. SP has the ability to learn a users true ID | 1 | 1 | 1 | 1 |
| 5. IdPs can only link attributes with the users permission | 1 | 1 | 1 | 1 |
| 6. SPs are able to pull additional attributes only with the user's permission | 1 | 1 | 1 | 1 |
| 7. Can use the standard HTTP/S Protocol | 1 | 1 | 1 | 1 |
| 8. Uses standard protocols, pref SAML | 1 | 1 | 1 | 1 |
| 9. Supports Multi Hop Proxying | 1 | 0 | 0 | 0 |
| 10.Supports signed Assertions | 1 | 1 | 1 | 1 |
| 11. Assertions signed by their Authoritative Sources | 0 | 1 | 1 | 1 |
| 12. What level of user interaction is required | Medium | High | High | Low |

Whilst none of the models can provide every requirement proposed in this paper, each model has its own strengths and weaknesses. Provider chaining is the only one of the four models to offer multi-hop proxying but the use of repackaged attribute assertions from potentially any IdP, which the SP may or may trust, could presents significant problems to some applications. SP mediated aggregation has a simple message flow but requires each SP to either have a list of all the user's IdPs to connect to (which places a high burden of trust on the SP) or the user must be asked for each IdP in turn. In both SP mediated and client based collection the user must authenticate at each IdP, so there is a high level of user interaction required in order to obtain the aggregated attribute set. In the latter model the smart client must be configured with a list of IdPs to query for attributes, tying users to a single configured client which may not be available in all circumstances e.g. when using a public PC or roaming. The Identity linking model, whilst not supporting multi-hop proxying, does have the lowest level of user interaction, only requires the user to authenticate once and does not tie the user to any SP or configured client as all the required links come from the IdPs or discovery service.

## 4. Future Work

In order to support the largest set of requirements that have been derived from our survey, we are in the process of defining aggregation protocols for the Identity linking model and intend to submit draft OASIS profiles for peer review in the near future. The proposal is to have three protocols, one for a user linking together her attributes at a Linking Service, which is a simplification of Liberty's discovery service [7]; one to allow the Linking Service to aggregate the attributes in a user's session, and a second to allow the SP to aggregate the attributes, based on referrals provided by the Linking Service. All three protocols will be standard extensions to SAMLv2

## Acknowledgements

We would like to thank the UK JISC for funding this research under the Shintau project.

## References

[1] Wang, L., Wijesekera, D., and Jajodia, S. A logic-based framework for attribute based access control, In *Proceedings of the ACM Workshop on Formal Methods in Security Engineering*, 2004, 45-55.
[2] See http://msdn2.microsoft.com/en-us/netframework/aa663320.aspx
[3] David Chadwick. “Authorisation using Attributes from Multiple Authorities” in Proceedings of WET-ICE 2006, June 2006, Manchester, UK
[4] Robinson, John-Paul. “MyVOCS: A Distributed Collaboration System”. Presentation available from http://www.stonesoup.org/Meetings/0609/vo.pres/robinson.pdf
[5] Scott Cantor. “Shibboleth Architecture, Protocols and Profiles”, Working Draft 10, September 2005, see http://shibboleth.internet2.edu/shibboleth-documents.html
[6] Jeff Hodges, Tom Wason (Eds). “Liberty ID-FF Architecture Overview” DRAFT Version 1.2-03, 14 April 2003
[7] Hodges, Jeff, Cahill, Conor (Eds). “Liberty ID-WSF Discovery Service Specification v2”. Liberty Alliance Project, 27 March 2006.

## Appendix 1. Questionnaire Recipients

Members of the Jericho Forum (http://www.opengroup.org/jericho/)
OGF OGSA Working Group list
OGF OGSA Authz WG list
Liberty Alliance group working on attributes
Sun's Identity and Access Management group
The XACML TC
JISC-MIDDLEWARE-DEVELOPMENT list
IDENTITY-PROJECT-PUBLIC JISC mailing list
Terena EMC2 mailing list
Shibboleth Dev list
gsmv@webapp.lab.ac.uab.edu
the OSIS list (osis-dev@netmesh.org)