The Trusted Attribute Aggregation Service (TAAS)

Providing an attribute aggregation layer for federated identity management

David W Chadwick
School of Computing
University of Kent
Canterbury, UK
d.w.chadwick@kent.ac.uk

George Inman
School of Computing
University of Kent
Canterbury, UK
gli3@kent.ac.uk

***Abstract*— We describe a web based federated identity management system loosely based on the user centric Windows CardSpace model. Unlike CardSpace that relies on a fat desktop client (the identity selector) in which the user can only select a single card per session, our model uses a standard web browser with a simple plugin that connects to a trusted attribute aggregation web service (TAAS). TAAS supports the aggregation of attributes from multiple identity providers (IdPs) and allows the user to select multiple single attribute "cards" in a session, which more accurately reflects real life in which users may present several plastic cards and self-asserted attributes in a single session. Privacy protection, user consent, and ease of use are critical success factors. Consequently TAAS does not know who the user is, the user consents by selecting the attributes she wants to release, and she only needs to authenticate to a single IdP even though attributes may be aggregated from multiple IdPs. The system does not limit the authentication mechanisms that can be used, and it protects the user from phishing attacks by malicious SPs.**

Keywords-attribute aggregation, identity management

I. INTRODUCTION

A user's digital identity can be stated as the set of attributes used to represent the user within a specific context. In the standard model for federated identity management systems (FIMS), such as Shibboleth [2] or Liberty Alliance [11], the context is the federation or circle of trust and the attributes are taken from the authentication and attribute assertions (or claims) released by the user's identity provider (IdP) to the service provider (SP). FIMS were often built under the assumption that a user would use a single institutional or corporate IdP for accessing each SP in the federation. As FIMS mature the size and scope of them become increasingly large e.g. in 2013 the UK Access Management federation had 940 members [3]. It is increasingly likely that a user will have several accounts at different IdPs within these federations. Furthermore, as the authorisation requirements of SPs become more complex, it is increasingly unlikely that one IdP can provide the full set of attributes that an SP requires. Consider purchasing a car parking permit online from a local council. The user may need to provide the following authorisation attributes:

- Proof of name and address, i.e. that they live in the local area
- Proof that they have a car
- Proof that they have a credit card to pay the fee.

These attributes will typically be asserted by different IdPs. In the UK, these are most likely to be: the Department of Works and Pensions, the Driver Vehicle Licensing Authority and a bank respectively.

Although the Information Card/Windows CardSpace model is no longer supported by Microsoft, nevertheless it had some excellent features in terms of identity management, usability and security. A good high level overview of CardSpace can be found in [1]. It assumed that a user has accounts at different IdPs, and that each IdP will issue claims containing different lists of her attributes. Each information card is a representation of a partial identity of a person's online digital identity and the full set of cards is a representation of the user's entire digital identity. Information cards can either be self-asserted or IdP asserted. From a usability perspective InfoCards provide a metaphor that is familiar to users and is reminiscent of the plastic cards that everyone carries around today in their wallets. From a security perspective, conventional phishing attacks are thwarted, since the user's identity selector redirects the browser to the genuine IdP, rather than a malicious SP redirecting the browser to a false IdP. Consequently we have kept these features.

However, CardSpace had some significant flaws. The WS-Trust exchanges [13] require the user to provide her authentication credentials to the Identity Selector. This severely limits the types of authentication method that can be utilised by the IdPs. CardSpace also made the same two fundamental mistakes as other FIM models by assuming that (a) the user need only select a single IdP (or card) in a session, and (b) this IdP will issue all the user's attributes that are needed for this SP. Contrast this to plastic cards, where users typically have lots of them issued by different IdPs, and each card typically holds only one (or very few) user attribute(s), i.e. the attribute the IdP is authoritative for, along with supporting information such as: the name of the user, the validity period of the card, a unique card identifier, a mechanism to authenticate the user (usually a signature or PIN, but could be a photograph as well), and details of the card issuer. Other contents such as holograms and chips are there to ensure the authenticity of the card and to stop forgeries. They do not provide additional attributes of the user. It is therefore not uncommon today for a user to provide several cards in a single transaction, along with self-asserted information. Thus as FIMS expand to Internet scale, users will need to aggregate their attributes or claims from multiple IdPs, as well as provide some self-asserted attributes. This is what TAAS provides.

The use of multiple IdPs has several advantages to users and SPs. A single IdP is no longer required to issue all of a user's attributes, which is an unrealistic assumption to make. Legal constraints and liabilities will make it very difficult for any IdP to make claims about a user's attributes for which it is not authoritative. For example, a credit card issuer would never make assertions about a user's driving ability, nor would a driver licensing centre make claims about credit worthiness. Rather each IdP will assert the attribute(s) for which it is authoritative and willing to bear the risk. This means that a user will need to pick from multiple IdPs the subset of their attributes that they wish to present to a SP, rather than passing their entire (sub)set in a single card issued by a single IdP. However this presents severe difficulties to today's IdPs, since no single IdP knows all the other IdPs at which a user has accounts, and users will probably want to keep it this way.

We have already published details about our privacy-preserving Linking Service for attribute aggregation [4], which allows users to perform a single login and then aggregate additional attributes from other linked IdPs based on a pre-set policy. A summary of the user trials can be read in [6]. This revealed that users want to *dynamically* choose which attributes from a specific IdP should be asserted and which shouldn't. This is what TAAS provides. So the contributions of this paper are: the description of a conceptual model for a privacy preserving dynamic trusted attribute aggregation service, and its implementation, which provides users will full fine grained control over the release of their attributes, both self-asserted and IdP asserted, without it knowing the identity of the user, which allows any authentication method to be used, prevents phishing attacks, and uses standard protocols and standard web browsers with a small plugin.

This rest of this paper is structured as follows. Section 2 describes the conceptual model. Section 3 describes our implementation using standard protocols. Section 4 concludes and says where further work is still needed.

II. CONCEPTUAL MODEL

Today's federations use a layered communications approach in which each layer utilises the services of the lower layer to provide additional value to the application layer. The bottom layer is the TCP/IP connectivity layer which provides end to end synchronous connections between two parties. The client server layer builds on this to provide user authentication between the user client and the server (an IdP). The federation layer builds on top of the client server layer to provide authentication between two previously unknown parties (the client and the SP). Our model proposes an additional layer on top of this, the attribute aggregation layer. This contains an attribute aggregator that is responsible for aggregating a user's self-asserted attributes with assertions from multiple IdPs, before presenting the combined set of attributes to the SP.

This model assumes that the user is the only person who knows about all her IdP accounts, and that she does not wish the other IdPs to know this information. We assume that some IdPs have pre-existing trust relationships with other IdPs and SPs but not that universal trust relationships exist between all entities. We do however require that an SP trusts all the IdPs that it receives attribute assertions (claims) from, and that the SP, the user and the IdPs trust our new attribute aggregation web service (called the Trusted Attribute Aggregation Service - TAAS) to confidentially hold the user's set of aggregated IdP accounts. The purpose of TAAS then, is to hold, in a privacy preserving way, the links to, and the attribute types held by, the user's different IdPs, as well as the user's self-asserted attributes. TAAS acts as the user's identity selector eliminating the need for a fat client on the user's desktop. As the IdPs link to TAAS they have no knowledge of any of the user's other IdP accounts. TAAS knows that some user has a set of linked IdP accounts, but it does not have any specific knowledge of who the user is, or the values of the attributes that are stored in each individual IdP account, apart from the user's self-asserted attributes. When linking an IdP account to TAAS the IdP releases only the types/names of the attributes and not the actual attribute values. Only the SP receives the attribute values, digitally signed and encrypted by their originating IdPs, as a result of attribute aggregation during the service provisioning phase.

The attribute aggregation protocol requires an IdP to release a user's attributes to the SP when TAAS requests it, without the user authenticating to the IdP. Consequently, TAAS needs to assure itself that the initiator of the current session is the IdP account holder and not an imposter. We solve this, along with the privacy requirement that TAAS does not know the identity of the user, by requiring the user to directly authenticate to each IdP during the linking phase, and for the IdP to provide a shared secret to TAAS which can subsequently be used to identify the user in future attribute aggregation sessions.

A. Link Registration

IdP accounts may be linked and configured at TAAS either before or during the service provisioning phase. Accounts are linked by the user authenticating to several IdPs in the same session. A user's initial TAAS account is created by navigating to TAAS, which acts as a conventional SP, and logging in using any IdP that has a trust relationship with TAAS. TAAS forwards the user to the IdP's authentication endpoint, which presents the user with an authentication dialog, allowing the user to authenticate with her credentials. After authentication, the IdP should ask the user which attribute types and values she wants to make available to TAAS for subsequent aggregation in service sessions. The IdP is then able to generate an attribute assertion for TAAS containing the user's chosen subset of her attribute types/names that the IdP is able to issue for the user. If the user selects a multi-valued attribute then the IdP should return both a type/name and pseudo-values to allow TAAS to refer to individual values during the selection phase, without TAAS learning the actual values. If no attribute assertion is returned it is assumed that the IdP will only be available for authentication and will not release any attributes for aggregation. Each attribute (pseudo-value) contained in the assertion will subsequently be displayed to the user as a separate card for selection during service provisioning.

The login request from TAAS to the IdP requests an authentication token containing a randomly generated but persistent identifier (PId) for the user. This PId will be stored by TAAS and subsequently used as a pair-wise secret between TAAS and the IdP in order to identify the user's account in all future communications between the two parties.

When TAAS receives a new (previously unknown) PId at login time, it creates a new entry for the user in its internal database. When TAAS receives an existing PId at login time, it retrieves the user's existing entry from its database and updates it to reflect the current set of attributes. For each linked IdP, TAAS stores: the PId of the user at this IdP, the set of attribute types/names and pseudo values that can be released by this IdP, and the level of assurance (see below). If the user wishes to link additional IdP accounts to her existing TAAS entry then she authenticates to another IdP and TAAS requests another set of information which it adds to the same user entry.

B. Level of Assurance

Different IdPs authenticate users in different ways and to different strengths. This is termed the Level of Assurance (LoA) [5]. It can be loosely thought of as how sure a relying party can be that the user is really who they say they are. This depends not only on the authentication method – which we term the Authentication LoA – but also on the initial vetting and registration process that the user underwent – which we term the Registration LoA. NIST [5] classifies a user's LoA at four levels, with level 4 being the strongest and level 1 being the weakest. A limitation of the NIST recommendation is that its LoA is a compound metric dependent on both the authentication method and the registration process. We believe that they are more useful if they are separate metrics, since IdPs may offer different authentication methods and a static registration procedure, or may alter the registration procedure that is used with the same authentication method. Thus we introduce a dynamically calculated Session LoA into the TAAS protocol. An IdP should set the Session LoA to the value of the Authentication LoA for the authentication method used by the user for this session, but with one proviso. No Session LoA can be higher than the Registration LoA used by the IdP when the user last registered with it.

When the user first registers with TAAS, the incoming Session LoA is stored as the user's Registration LoA with TAAS. During the service provision phase TAAS will only use linked IdPs whose Registration LoA's are higher than or equal to the current Session LoA, determined by the authenticating IdP. This prevents the user from creating links with low Registration LoAs and using them at higher Session LoA's, since this may give an escalation of privileges. A user can create links at high Registration LoAs and use them at lower Session LoAs, since this is not a security risk.

When a linked IdP receives a service request for attributes from TAAS, it extracts the Session LoA from the authentication assertion and compares this to its local Registration LoA. If the Session LoA is less than or equal to the latter then the IdP will release additional attributes to the SP (via TAAS), otherwise if the Session LoA is higher than the Registration LoA then the IdP will not release any attributes to the SP.

C. Service Provision Phase

We introduce a thin client - the TAAS discovery plugin - for the user's browser. This plugin is activated when TAAS login is requested by a SP. It is used to discover the user's TAAS, redirect the user to this TAAS and return the multiple IdP assertions from TAAS to the SP. When a user navigates to a TAAS enabled SP's restricted page it should explain that user login is required and should describe its security policy, in terms of which attributes are needed to

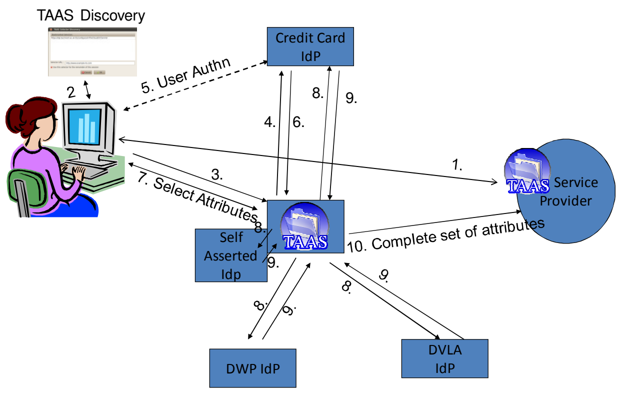

Figure 1. TAAS Protocol Flow

access the protected content, and how strong the authentication should be. This makes the user aware of what is required, providing greater transparency than with many of today's federated IdPs. It allows the user to decide which TAAS and linked IdPs to use and consent to providing the necessary attributes.

The SP's page should contain a TAAS icon (or other button) for the user to select, which has an embedded MIME object containing the SP's security policy (see below). When the browser attempts to process this embedded object it triggers the TAAS discovery plugin registered as the MIME handler. The plugin displays the TAAS discovery screen which allows the user to choose her preferred TAAS, thus thwarting phishing attacks since a malicious SP is not able to redirect the user to a TAAS of its own choosing. The URL(s) of the user's preferred TAAS(s) can be added to the plugin in two ways. The user can bookmark a site as a TAAS by saving its home page URL in a bespoke bookmark folder accessible by the plugin. Alternatively the user can dynamically add a new TAAS when the plugin is invoked by entering its URL directly into the plugin. This allows users to utilize TAAS from any device without having to pre-store bookmarks of their preferred TAAS(s) beforehand. Thus our system can be used on any public or Internet café computer without releasing personal information to other users.

When the user has selected her TAAS the plugin establishes a TLS connection with TAAS to protect all future communications between the browser and TAAS. The plugin constructs a HTTP POST message containing the SP's security policy for the protected resource and redirects the user to the TAAS URL. The receipt of the SP's security policy is treated as a service provision request for the required attributes to be aggregated and delivered to the SP. TAAS parses the security policy and extracts the set(s) of attributes and their associated issuers.

If the user has already established a prior session with TAAS from this browser, the browser's cache may contain a valid TAAS cookie containing the user's account identifier at TAAS, and a valid single sign on (SSO) cookie that may allow TAAS to authenticate the user transparently with the IdP. If not, TAAS will not be able to access the user's account and will need to discover one of the user's IdPs e.g. by using a Where Are You From (WAYF) service to display the list of all the IdPs that it has trust relationships with. The user chooses one of these IdPs and is redirected there for authentication. TAAS asks the

IdP to generate an authentication assertion, containing a randomly generated transient identifier, which acts as an SSO token valid at any entity that trusts this IdP, and a "referral" assertion containing the pre-linked PId which points to the user's account at TAAS. The user logs in to the IdP, and the IDP returns both assertions to TAAS.

TAAS is now able to access the user's account and display the set of "cards" that match the SP's policy. Each card icon represents one part of the SP's security policy. Multiple sets of cards will be displayed if the SP provides several alternative policies (see later). Cards are marked as either required or optional, depending upon the SP's policy. All required cards must be selected before aggregation can take place. We envision the optional cards will represent self-asserted attributes required by the SP for communication or marketing purposes rather than authorization. The user clicks on the card icon and this displays an attribute selection overlay containing all the user's previously linked attributes that match this part of the SP's policy. Each attribute is shown as a card with its name, issuer and pseudo-value (as provided to TAAS by the IdP) and the user chooses the attribute/card she wishes to send to the SP. For example, if the SP required a credit card from a bank, when the user clicks on the card icon, the overlay will display each credit card type from each bank that the user had previously linked with TAAS, allowing the user to dynamically choose her preferred credit card attribute for this transaction.

If no previously linked attributes match the SP's security policy, then when the user clicks on the card icon, TAAS asks the user to dynamically link a new IdP account to her existing TAAS account. If the user agrees, a new discovery/WAYF screen is shown that only displays those IdPs that are trusted by the SP to issue the required attribute. Once the user has authenticated to one of these IdPs, the new IdP account details are added to the user's existing TAAS entry, before the attribute selection card overlay is reshown to the user. The user is now able to see the newly linked attribute and select it, in order to fulfill this part of the SP's security policy. This allows users to dynamically link all their needed attributes during the service provisioning phase without any prior setup.

The user selects an attribute by double clicking on its card and the overlay is removed. The card icon is now shown with a large green tick on it. Once all the cards have been ticked the user can choose one of three options:

1. Click the Submit button, in which case the selected attributes will be aggregated from the IdPs and sent to the SP (but the selections will not be remembered).
2. Click the Save and Submit button, in which case the selected attributes will be remembered in TAAS's database before aggregation commences. In this case, if the user attempts to access the same resource at the same SP again, the last set of selected attributes will be pre-selected and the card icons displayed with large green ticks.
3. Click the "Don't Bother Me Again" tickbox and then click the Save and Submit button. This stores the selected attributes along with a flag, before aggregation commences. If the user accesses the same resource at the same SP again, TAAS will transparently aggregate the same set of attributes again, without showing the attribute selection page to the user. This equates to the "one-click" functionality of Amazon.

After the user clicks one of the submit buttons, TAAS will query each of the chosen IdPs for the user's attributes, as well as its database for any pre-stored self-asserted attributes. An IdP query comprises: an attribute query requesting a subset of the SP's requested attributes (as chosen by the user), an encrypted referral pointing to the user's account at the IdP and the original authentication assertion. The recipient IdP uses the authentication assertion to determine whether it trusts the initial act of authentication by the authenticating IdP. If the authenticating IdP is not trusted then the recipient IdP should return an error to TAAS. If the IdP does trust the authenticating IdP then it generates an attribute assertion containing the user's attributes and encrypts this assertion to the SP (which it must also trust to privacy protect the user's attributes). The user is identified in this attribute assertion with the random transient identifier from the authentication assertion, so that all the assertions from all the IdPs will contain the same user identifier. The attribute assertion is returned to TAAS, which stores it until all the queried IdPs have replied.

Self-issued attributes are stored in TAAS's database by the user selecting the Manage My Personal Details tab at any time. This allows the user to add any identity attribute types and values that she wishes, whether fictitious or not. Since they are self-issued TAAS does not care what the contents are. When a self-issued attribute is selected by the user to fulfil an SP's policy, TAAS creates a new attribute assertion encrypted to the requesting SP with TAAS as the issuer. This newly created assertion can then be added to the set of attribute assertions collected from the IdPs.

Once all the attribute assertions have been collected by TAAS it generates a POST response to the endpoint specified in the SP's security policy, which the browser uses to redirect it to the SP. This response contains the authentication assertion, its own encrypted attribute assertion (for self-asserted attributes) and each of the encrypted attribute assertions returned from the IdPs. Consequently the SP receives a set of assertions containing a single authentication token and multiple attribute assertions which all contain the same random transient identifier. Since the SP trusts all the authoritative sources and TAAS, and can verify that they have all issued the assertions from their signatures, it can be assured that the same user possesses all of the returned attributes, and has been successfully authenticated.

D. Service Provider's Security Policy

The SP's security policy is an XML structure stored within a MIME object. Conceptually this structure offers a similar request structure to a SAMLv2 Attribute Request message but it has been expanded to encompass requesting attributes from multiple IdPs. Conceptually this policy consists of four pieces of information:

- The identifier of the SP and its associated public key with which authentication and attribute claims can be encrypted.
- The endpoint to which claims for the protected resource should be submitted.
- A list of IdPs which the SP trusts to authenticate the user and their minimum acceptable Authentication LoAs.
- Its authorisation policy, consisting of one or more attribute sets each of which describe a collection of

attributes. An attribute comprises the type/name of the attribute and a list of all the IdPs that the SP trusts to issue values for it.

Authorisation policies are specified in either conjunctive normal form (CNF) or disjunctive normal form (DNF):

- In CNF each attribute set consists of one of more attributes of which exactly one attribute must be chosen. The chosen attributes are then combined together to form the complete set required to access the protected resource. An example of this is the car parking permit policy described in the Introduction.
- In DNF all the attributes in a single attribute set must be provided to access the protected resource at the SP, but multiple alternative sets can be defined. An example of this is: to download a paper from an online journal, a user may provide either a credit card attribute from a bank, or a current journal membership number, or a proof of faculty membership from a university.

CNF policies allow the SP to specify groups of similar attributes that are equivalent, whereas DNF policies allow completely different alternative sets of attributes to be specified. The SP's policy allows TAAS to transparently filter out untrusted IdPs from the aggregation process and ensures that only those attributes that fulfil the SP's authorisation requirements will be aggregated.

III. Implementation

Our conceptual model has been implemented using the Security Assertions Markup Language (SAML) v2 protocol and customised HTTPS POST messages for interactions between the SP and TAAS. We use the SAML Identity Assurance profile [7] to pass the Session LoA between components.

A. *Link Registration Protocol*

TAAS is assumed to have received the SAML metadata [12] of the IdPs it trusts prior to user linking. It uses these to determine the SAMLv2 authentication endpoints, and to construct standard SAMLv2 <samlp:AuthnRequest> messages to these endpoints. This message requests that a PId be returned as the Subject of the <samlp:Response>. It uses the AttributeConsumer Index attribute to specify that all available attributes should be returned in the response. To ensure that the IdP always returns a PId:

- the Format attribute of the <NameIDPolicy> is set to "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent",
- the allowCreate attribute of the <NameIDPolicy> is set to true, which allows the IdP to create a PId if none already exists.

In response, the IdP constructs a <samlp:Response> message containing a single authentication assertion consisting of two statements, an Authentication statement element and an Attribute statement element. The Authentication statement describes the actual act of authentication performed at the IdP and the attribute statement contains the set of SAML attribute types/names that can be aggregated from this IdP.

B. *Service Provision Protocols*

The protocol mappings for attribute aggregation use a combination of existing SAMLv2 protocols and HTTPS POST operations. The former is used between TAAS and the IdPs and the latter between the SP and TAAS via the browser. The SAML protocols encode referrals as Liberty Alliance ID-WSF Endpoint References (EPRs) according to the EPR generation rules defined in Section 4.2 of [8]. The EPR's <sec:Token> element contains a SAMLv2 bearer assertion with the encrypted PId of the user as the assertion's Subject element. This allows the recipient to determine which subject is being referred to.

C. *SP to TAAS interactions*

When the user clicks on the TAAS icon, the browser detects the presence of a specific embedded MIME object in the HTML page and activates the TAAS discovery browser plugin. The plugin allows the user to choose her preferred TAAS and it constructs a new HTTP POST message to the chosen TAAS page. This POST message contains the SP's security policy embedded in the MIME type as a POST parameter named spPolicy. TAAS parses the XML security policy and, if the user is not already logged in, uses it to display a WAYF page that matches the SP's authentication requirements.

Once attribute aggregation is complete TAAS has a single SAML authentication assertion and zero or more SAML attribute assertions to relay to the SP. These attribute assertions are all signed by their issuers and encrypted to the SP. When the user presses a submit button, TAAS creates a new SAMLv2 Assertion, with a single attribute statement containing each of the aggregated assertions as separate attribute values. This assertion is signed by TAAS and returned to the SP's endpoint defined in its security policy, using a TLS encrypted HTTP POST message that requests the browser to return control to the SP. The SP can decrypt the message to access the attributes and authentication details and authorise the user.

D. *User Authenticating to an IdP*

When the user needs to authenticate to TAAS, TAAS constructs a <samlp:AuthnRequest> message requesting that the IdP returns a signed SAMLv2 Authentication SSO token and a separate encrypted SAMLv2 bearer attribute assertion valid at itself. The authentication SSO token should be unencrypted, contain a transient/random ID for the user, and be valid at any recipient. The bearer token must contain the LA ID-WSF EPR as a subject attribute, with the PId set to allow TAAS to identify the authenticated user. This is accomplished by setting the AttributeConsumingServiceIndex attribute to request that the EPR attribute is returned, which also informs the IdP that the attribute aggregation profile has been selected. After the user has authenticated to the IdP, by any method it supports, it constructs the authentication assertion and sets the AuthnContext element to contain the session LoA for the authentication method that was used (subject to it having a maximum value of the Registration LoA). These assertions are then placed in a <samlp:Response> message and returned to TAAS.

TAAS decrypts the referral contained in the wsse:Security element of the EPR, using its private key, to obtain the PId of the user. It looks up the user in its database, and filters the user's linked accounts against the SP's security policy. It then displays the attribute selection screen to the user.

E. Attribute Aggregation with the IdPs

Normal HTTPS request/response messages are used between the browser and TAAS to enable the user to select which attributes from which IdPs she wishes to aggregate together.

Once the set of IdPs has been determined, TAAS constructs a referral to each IdP by creating a SAMLv2 Bearer assertion containing an EPR with the encrypted PId of the user that is shared between itself and the IdP. TAAS then constructs a SAMLv2 attribute query message to each IdP, using the SAMLv2.0 Attribute query profile from SAML Core [9], requesting the subset of attributes that the user has chosen to be returned from this IdP. The SOAP envelope containing the AttributeQuery message has a <samlp:AttributeQuery> as its body element and a <wsse:Security> [10] element in its header, which comprises the newly constructed SAMLv2 Bearer assertion and the original authentication assertion from the authenticating IdP. The subject of these attribute queries is the transient/random identifier extracted from the initial authentication assertion. The IdP identifies the user's account by decrypting the PId in the SAMLv2 Bearer assertion stored in the <wsse:Security> header and determines whether or not it trusts the initial act of authentication by examining the signature of the authentication assertion in the same <wsse:Security> header. The IdP can differentiate between the referral assertion and the authentication assertion as only the ID in the referral assertion will match a stored PId in the IdP's system, and this assertion will directly reference the authentication assertion. It then uses the transient/random subject identifier from the authentication assertion as the subject ID in the newly generated attribute assertion.

As it is TAAS that is querying the IdP directly rather than the SP, it is necessary to include an AssertionConsumerServiceURL attribute in the AttributeQuery message that specifies the SP as the entity to which the resulting attribute assertion should be encrypted (Section 3.4.1 of [9]). The encrypted attribute assertion is then inserted into a <samlp:Response> and returned to the querying TAAS.

TAAS waits until each queried IdP has responded and collates the combined set of attribute assertions into a single SAMLv2 Assertion. This contains a single attribute statement containing each of the aggregated assertions as separate attribute values. Any self-asserted attributes stored in TAAS's database form an additional assertion value. The single assertion is signed by TAAS and returned to the SP's endpoint defined in its security policy by using a TLS encrypted HTTP POST message that is returned to the browser, which posts it to the SP.

IV. Conclusions and Future Work

Whilst the CardSpace model had some notable security and usability properties, it also had some significant flaws. We have combined the best user-centric features from CardSpace with the existing SAMLv2 security framework to provide a system that is better than the sum of its parts. The resulting TAAS system provides users with a more user friendly portable approach to identity management, which gives them fuller control and consent over the selection and release of their attributes, whilst protecting them from phishing attacks. TAAS also goes beyond the capabilities of either previous system by providing built in support for attribute aggregation. We believe this will become an increasingly important access control requirement as applications become more demanding and security conscious, where the present model of expecting a single IdP to hold all a user's attributes will no longer be sufficient.

We have implemented the work described in this paper as a fully featured demonstrator[1] and released the code as open source as part of the Open PERMIS project[2]. Future work could be to provide the TAAS functionality entirely within the browser, as an alternative to using an external trusted server. Work is still required to define a standard ontology for both attributes and IdPs, so that an SP's security policy can refer to generic entities rather than to individually named attributes and IdPs, which could get lengthy if there are many trusted entities of the same type.

Acknowledgment

The research leading to these results has received funding from the UK JISC and the EC's FP7 programme under grant agreement n° 216287 (TAS³ - Trusted Architecture for Securely Shared Services).

References

[1] David Chappell. "Introducing Windows CardSpace". MSDN. April 2006. Available from http://msdn.microsoft.com/en-us/library/aa480189.aspx

[2] R. L. "Bob" Morgan, Scott Cantor, Steven Carmody, Walter Hoehn, and Ken Klingenstein. "Federated Security: The Shibboleth Approach". Educause Quarterly. Volume 27, Number 4, 2004

[3] http://www.ukfederation.org.uk/content/Documents/MemberList

[4] David W. Chadwick, George Inman, Nate Klingenstein "A Conceptual Model for Attribute Aggregation". Future Generation Computer Systems. Volume 26, Issue 7, July 2010, Pages 1043-1052.

[5] William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, Emad A. Nabbus. "Electronic Authentication Guideline", NIST Special Publication 800-63-1, Dec 2011.

[6] John Watt, Richard O Sinnott, George Inman, David Chadwick. "Federated Authentication and Authorisation in the Social Science Domain". Sixth International Conference on Availability, Reliability and Security (ARES), 22-26 Aug 2011, pp541-548

[7] OASIS. "SAML V2.0 Identity Assurance Profiles Version 1.0" Committee Specification 01, 5 November 2010.

[8] Hodges, J. Aarts, R. Madsen, P. and Cantor, S.(Editors). "Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification v2.0". Liberty Alliance Project.

[9] OASIS. "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard, 15 March 2005

[10] Hodges, J. Kemp, J. Aarts, R. Whitehead, G. Madsen, P. "Liberty ID-WSF SOAP Binding Specification v2.0" Liberty Alliance Project.

[11] Liberty Alliance Project. "Liberty ID-WSF Web Services Framework Overview" Version: 2.0" Available from http://www.projectliberty.org/specifications__1

[12] OASIS. "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0". OASIS Standard, 15 March 2005

[13] OASIS, "WS-Trust 1.3", OASIS Standard, 19 March 2007

[1] TAAS Demo. http://sec.cs.kent.ac.uk/demos/taas.html

[2] http://www.openpermis.org and http://sec.cs.kent.ac.uk/permis.