Expression of Interest – Improving Identity Management on the Internet

David W Chadwick, George Inman, Kristy Siu, University of Kent

1. Introduction and Interest in the Workshop

We have been doing research in identity management for several years, particularly in the area of attribute aggregation. The reason for this is simple – everyone has identity and entitlement attributes issued by multiple attribute authorities (AAs). No-one has all their attributes issued by a single Identity Provider (IdP). Yet all of today's federated identity management systems make the same fundamentally wrong assumption – that the authenticating IdP will always provide all of the user's attributes to the relying party (RP)/service provider (SP). This may be true some of the time, but certainly not all of the time. Consequently we have taken a different stance, namely, that the user wants to authenticate just once to the RP/SP, and then be able to provide his/her attributes from a variety of different attribute authorities as required by the SP e.g. her credit card from her bank, her role from her employer, her discount entitlement from a membership club and her postal address herself. The latter is just as important as the former. Users should be able to assert their own attributes as well as allowing trusted AAs to assert them on their behalf. User consent is also important. Users should be able to consent to their attributes being released each time they are released to an SP. Privacy is another important issue. A user's attributes should not be leaked to outsiders, and users should not have to reveal their true identity unless they choose to. They should be able to remain anonymous (or pseudonymous) and still obtain a service e.g. to download a paper from ACM's digital library I should only need to prove that I am a member of a university that has subscribed for the downloads. If I want to download other papers that have not been subscribed for by my university, I should then only need to prove I have a credit card to pay for this, and obtain the download, without revealing who I am (or my credit card number). So users should be able to selectively reveal their attributes as they request different services from a SP, and this process should be repeatable within a single session. Security is also important. SPs may need to have strong assurance about particular identity attributes. Thus levels of assurance should be built into the model. Above all, usability is a critical success factor. If users cannot easily use the system, then they will not.

All of the above can now be achieved by a simple browser plug in that we have developed, using existing standard protocols and a new Internet based trusted third party which we call the Trusted Attribute Aggregation Service (TAAS).

In developing TAAS we have utilised the latest state of the art systems and standard protocols, and borrowed good ideas already developed elsewhere. We do not pretend to have developed everything ourselves. So in our solution you will see technologies and techniques used previously to good effect by SAML, Shibboleth, Liberty Alliance and CardSpace, as well as a few innovations that we have developed ourselves.

Space restricts the amount of detail we can present in the position paper, but the overview and our previous publications should provide sufficient details for you to appreciate how our ideas have developed over recent years.

2. The Trusted Attribute Aggregation System

2.1 Trusted Attribute Aggregation Service (TAAS)

The Trusted Attribute Aggregation Service (TAAS) is a new trusted third party which holds privacy protected details about a user's different accounts at different IdPs. Federations (circles of trust)

may contain as many TAASs as they wish. The user manages her TAAS account and includes as much or as little information in it as she wishes. A user may have multiple TAAS accounts from different providers that both she and different federations trust. The information stored in TAAS by a user comprises the following:

- for each IdP she has introduced to TAAS:
  - the name of the IdP,
  - the LoA used by the IdP to authenticate the user,
  - the persistent identifier (PId) generated by the IdP to refer to this user at this TAAS i.e. the PId is a unique pairwise link as used in standard SAML protocols today;
  - a subset of the attribute types/names that this IdP can assert for this user (note that the attribute values are never known to TAAS);
- an optional set of self-asserted attributes (types and values) such as different names, addresses, telephone numbers etc. which may be fictitious or genuine. The user can add as many or as few of these as she wishes, depending upon the various requirements of different SPs.

A user introduces an IdP to TAAS using the existing SAMLv2 protocol exchange, in which TAAS acts as an SP. TAAS requires the user to login to the IdP and for the IdP to return as many attribute types/names as the user wishes to be aggregated by TAAS and its local policy allows. We supplement this exchange with a requirement for the IdP to return the LoA which it used to authenticate the user and this is stored by TAAS as above.

2.2 Browser Plug In

We register a new MIME type, say TAAS, and have a new browser plugin module that is the registered handler for this MIME type. The content of this MIME type is the SP's security policy. The SP's security policy can be hidden behind a new TAAS icon on the SP's page. When the user clicks on this icon to get access to protected resource at the SP, the security policy is sent to the browser, which causes it to invoke the new browser plugin module.

The purpose of the plug in module is to ask the user which TAAS she wants to use to process the SP's security policy and to redirect the browser there. This stops phishing attacks, which current SAMLv2 systems are prone to, in which the SP redirects the user to a masquerading IdP which then captures the user's login credentials. In our system, the plugin invites the user either to enter the URL of their favourite TAAS, or to pick the URL from the set of TAAS bookmarks which the user has previously saved in his browser. This feature supports mobile users with smart phones as well as travelling users in Internet cafes.

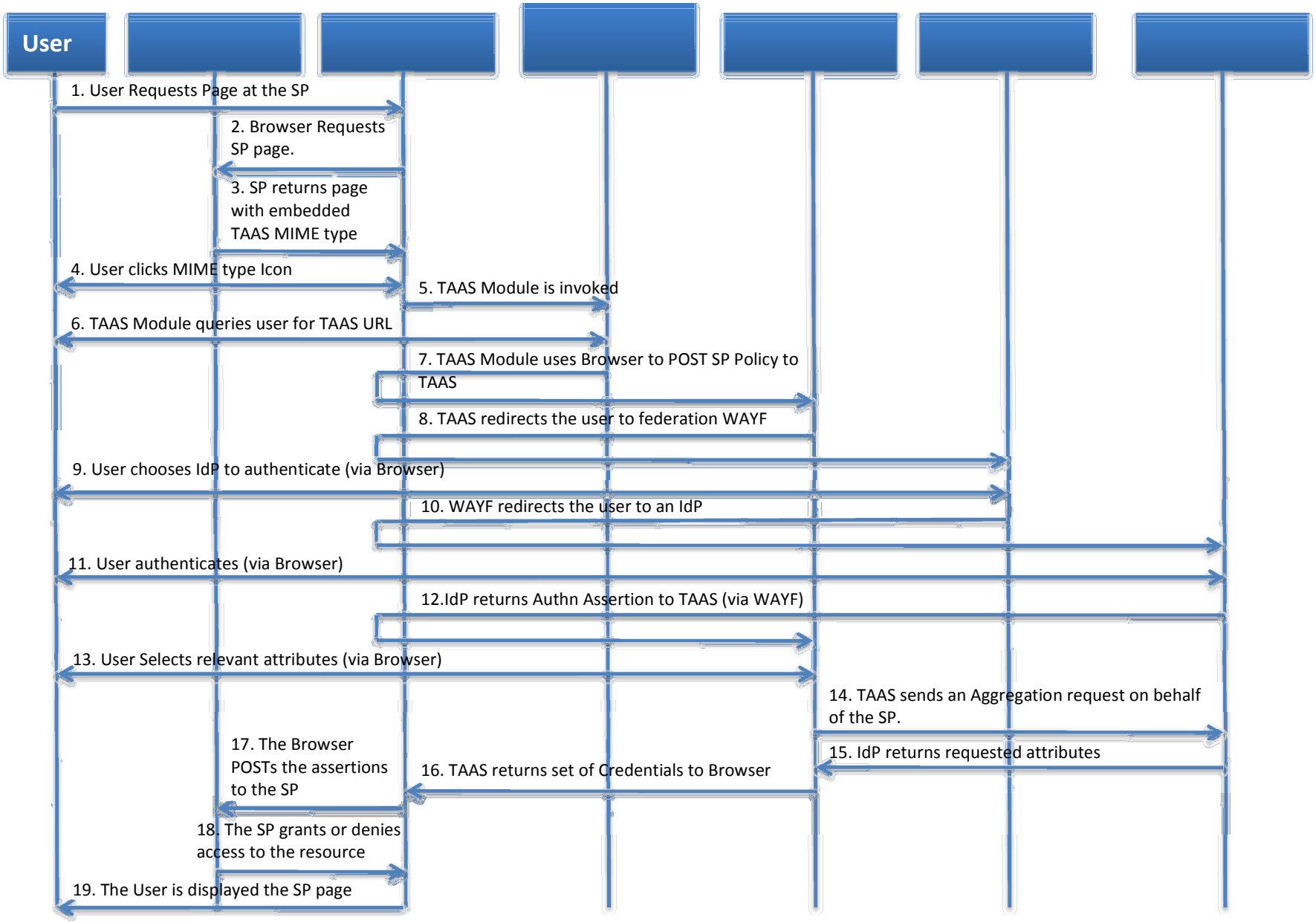
2.3 SP's Security Policy

This security policy contains details of the set of credentials that is required to access a protected page, along with the URL of a page to which these credentials should be submitted before the protected page can be displayed. The SP's attribute requirements are specified using either conjunctive or disjunctive normal form, since this caters for all possible combinations. Attributes can be flagged as being mandatory or optional. The returned credentials will comprise an authentication assertion with the minimum required Level of Assurance (LoA), and a set of attribute assertions issued by different issuers each with their own LoA.

2.3 Protocol Exchange

The actors in the protocol exchange are: the user, the user's browser, the SP, the TAAS browser plugin module, the TAAS service, an optional federation Where Are You From service, an authenticating IdP and one or more attribute providing IdPs. An overview of the protocol exchange is given in the diagram below, with fuller descriptions of each step later.

Figure 1. Protocol Flow



1. The user requests a protected page from his browser.
2. The browser performs a HTTP GET request for the page from the SP's server.
3. The SP returns a HTTP Response page containing an icon with a hidden TAAS MIME type that contains the SP's security policy.
4. The user clicks on the TAAS icon to start the authorisation process.
5. When the TAAS icon is clicked the MIME type is no longer hidden and the Browser attempts to process it. As the TAAS browser plugin module is the registered handler for this MIME type it is invoked.
6. The plugin module displays a new window to the user, showing the set of bookmarks containing her already saved TAASs, plus a blank space to enter a new URL. The user either enters a new URL or clicks on the bookmark of an existing one, and the window returns the value to the TAAS plugin module.
7. The TAAS plugin module returns the value to the browser which processes the URL and sends a HTTP POST message to the URL. This message contains the SP's security policy formatted as POST Parameters.
8. When the TAAS receives a request it checks to see if the user has already been authenticated in this session because a SSO cookie is present in the request. If it is, it decrypts the SSO cookie and extracts the user's persistent identifier (PID) from it and goes to step 13. If the user has not been authenticated then it parses the SP's security policy and extracts the required authentication LoA. It places the SP's policy in a cookie which will be stored in the user's browser and returned to it in step 12. The user is then redirected to the federation WAYF service along with a <samlp:AuthnRequest> that requests that the user be authenticated to the required LoA and that the response contains an Authentication assertion with the PID known to the TAAS, and containing a federation wide authentication assertion with a transient/random ID (RID).
9. The WAYF (enhanced to support LoAs) displays a filtered (to the required LoA) IdP selection screen to the user and she selects the IdP she wishes to authenticate with.
10. The WAYF then redirects the original <samlp:AuthnRequest> message to the IdP.
11. The IdP processes the authentication request and asks the user to authenticate at or above the required LoA.
12. The IdP returns a <samlp:Response> to the TAAS via the browser. This response contains a federation wide authentication assertion containing a RID and an authentication statement for the TAAS containing a PID. The PID is extracted. The response also contains the browser cookie containing the SP's security policy.
13. The PID is looked up in the TAAS's user database to see if an account already exists. If the user account is found, it means that this internally known user has now been authenticated. The user's account contains a list of attribute types and the IdPs which are the issuers of them for this user, along with any self-asserted attributes that the user has chosen to store there.
14. The SP's policy is filtered against the user's aggregated attribute types. All optional policy elements that the user cannot fulfil are discarded. If there are any mandatory elements that the user does not have aggregated attributes to match, or they are at too low an LoA, then the user is shown the account management screen, as she cannot obtain a service at the SP until she has aggregated these attributes at a sufficiently high level of assurance. Assuming the user has sufficient attributes to match the SP's policy, the user is displayed a web page that shows all the SP's attribute requirements, each requirement as one icon. Clicking on each icon reveals the user's set of aggregated attributes that match it, e.g. the SP may require a credit card attribute and the user may have several credit cards (as in Figure 2 below). The user then selects one of her attributes to match this policy requirement, and then moves onto the next requirement. After each requirement has been fulfilled, the icon displays the chosen attribute with a tick (as for the chosen airline frequent flyer card in Figure 2 below). Once all icons are ticked, the user may submit her selection via HTTP POST to the TAAS.

15. The TAAS queries each of the user's IdPs for the selected attributes using a <samlp:AttributeQuery> containing the RID as the identifier of the user. The SOAP header also contains a Liberty Alliance IDWSF Security assertion containing the encrypted PID of the user at that IdP, and the federation wide authentication assertion containing the RID.
16. The IdP decides if it trusts the authentication assertion, and assuming it does, it decrypts the PID, looks up the user in its database, selects the required attributes that match the requested ones, and returns them in a SAML attribute assertion bound to the RID and encrypted for the SP.
17. The TAAS aggregates the returned attribute assertions and encapsulates them in a single signed assertion along with the federation wide authentication assertion. Each aggregated assertion is signed by its authoritative IdP and identifies the user by the same RID.
18. The TAAS checks the SP's security policy for the specified page to return the credentials to and returns them to the browser in a FORM POST message to the SP. It also returns an encrypted SSO authentication cookie containing details of the authentication session and the user's internal account (PID).
19. The Browser POSTs the credentials to the SP's authorisation page. The SP processes the assertions and authorises the user.
20. If the user is granted access then the originally requested page is sent to the browser as a result of the POST.
21. The user is displayed the protected page.
22. If the user subsequently tries to access another page which requires further authorisation, the SP can automatically redirect the user to the TAAS she has chosen for this session, along with its new security policy, and the TAAS can filter this policy against the user's attributes and display another attribute picking list to the user in a repeat of the above process.

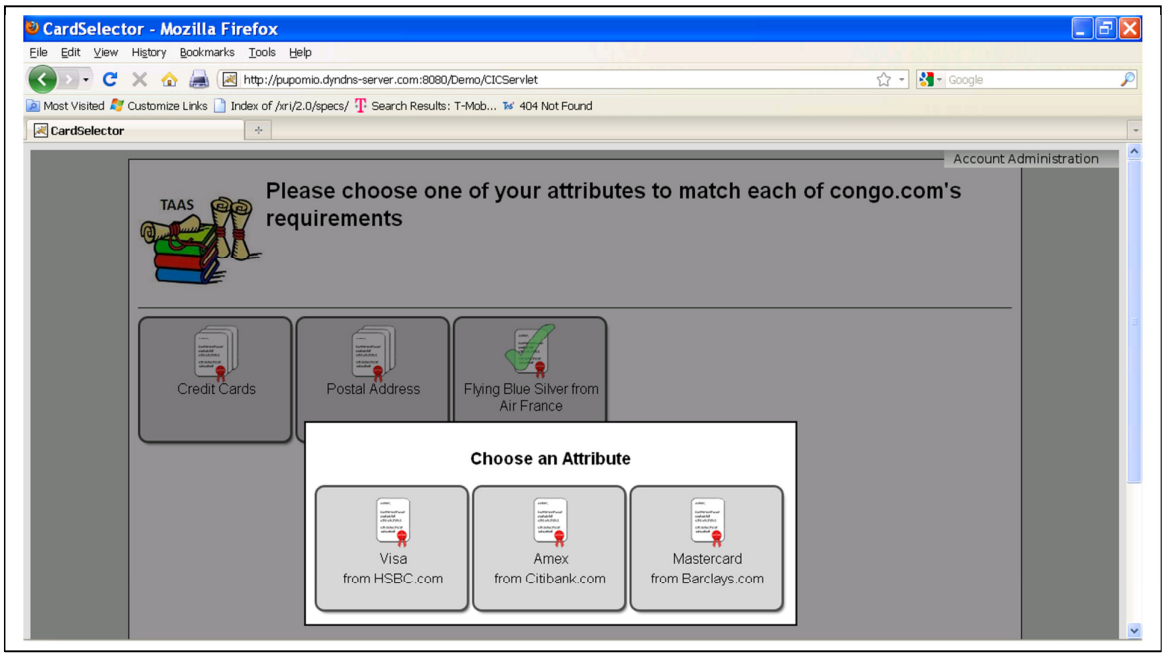


Figure 2. The Attribute Selection Page, showing that the SP congo.com requires 3 attributes (a credit card, a postal address and a frequent flyer card) and the user is in the middle of selecting one of her 3 credit cards.