
# Liberty Alliance Project Whitepaper: Personal Identity

March 23, 2006

## Executive Overview

Much of the discussion around "User-centric identity" positions it as something that will happen in the future, when in fact, identity management solutions based on Liberty Alliance standards make this possible today. User-centric identity involves users in the management of their personal information (with all the advantages and risks that this implies) and how that information is used, rather than to presume that an enterprise or commercial entity holds all the power.

The open identity protocols of the Liberty Alliance have built-in user consent and privacy features, which are designed to work with a wide variety of network devices. In addition, the Liberty model works equally well with human users and the machine-to-machine communications involved in service-oriented architectures.

Liberty-based identity management solutions allow enterprises to control user access to their resources and also allow users to control access to their personally identifiable attributes. Over 70 products have passed Liberty interoperability testing, which provides customers with a choice in the technologies they choose to deploy, with the confidence that they adhere to the same standards.

This document discusses the methods provided by the Liberty ID-FF and ID-WSF specifications for the making and verification of identity claims.

# Introduction

"On the Internet, nobody knows you're a dog" - *Peter Steiner, New Yorker*

If I were having my first casual conversation with you, and told you that my name is John, and I'm (only slightly) over twenty-one, you'd probably believe me. You'd have very few reasons not to. Say you met me again a few months later, and I told you that I'm originally from the United Kingdom, and that I like to play what is known in that country as "football" If you have a good memory, you might remember my face, and associate my latest statements (preference for football and nationality) with those I made previously (my name and age).

If, on the other hand, I'm sitting at my computer and I browse to your website, how do you know it's me, John? What if I fill out a form on your site and tell you that I'm only twenty years old, and that the football I like is actually of the "pointy-ball" variety popular in the United States? How about if I come to the same website again in a few months - how does the website know it's John again?

What if I ask you to lend me a lot of money, either in person, or via an email? You'd probably want to know a bit more about me than my name, nationality and hobbies!

In the "offline" world, where we can (sometimes) see each other, we often make fairly quick decisions to trust each other, particularly when it comes to sharing basic

information. But when the value of this information increases (when I ask you for a lot of money for example) or when we can't see each other (on the Internet) such decisions should be made with a little more care, as there are fewer cues to aid anyone relying on that trust. In such cases, it is quite common for someone who does not know you to ask to see some form of identifying information that backs up your claim. For example, someone selling you a beer might ask to see your government-issued driving license in order to verify that you are in fact older than the legal drinking age. This is both because the seller does not know you well enough to be sure that you will tell him the truth, and also because his liability in selling a beer to a minor can be quite high - he might go to prison.[1]

Technology has been invented to solve problems associated with this kind of trust. For example, the Kerberos (http://web.mit.edu/kerberos/www/) network authentication protocol allows one computer software application to authenticate to another securely over a network, and for one computer application to create secure "tickets" for the other which can then be used to authenticate to a third application. This is similar to the driving license situation noted above, where one entity (the barkeep selling you a beer) relies on an *identity claim* (your age) issued by a third party *authority* (the government, in the form of the DMV). Rather than the barkeep relying on my word alone, he is relying on a third-party *assertion* that backs up my word. He relies on this assertion because he can expect that the DMV has itself verified an individual's claim that she is of a certain age, perhaps by that individual showing the DMV her government-issued birth certificate. One issuer of identity claims (the DMV) trusts the claims of another issuer of identity claims (the registrar of births) forming a chain of trust. The person actually making the identity claim (who wishes only to buy a single beer!) is trusted only indirectly by the barkeep, based on that person's relationship with the registrar of births and the trusted relationship of the registrar to the DMV!

It is not only governments that make such identity claims on behalf of their citizens. For example, your Internet service provider (ISP) might offer you free access to a news service provided by some other website. This news website will need to know that users entitled to such free access are in fact who they say they are. In such a case, the website might depend on an identity claim issued by the ISP when one of its users first authenticates to the ISP. This illustrates *single sign-on*, where by authenticating to my ISP, I can now also get access to a partner of my ISP. Furthermore, my ISP is now willing to assert on my behalf to the news website that I am one of the ISPs users.

The Liberty Alliance Identity Federation specifications (http://www.projectliberty.org/resources/specifications.php#box1, now also part of the OASIS SAML 2 suite) support the above notions – that a user can obtain access to multiple websites via single sign-on, and that an *identity provider* (IdP) may make an

[1] A driving license is issued as a claim that the holder of the license has been tested, and is therefore licensed to drive. However, this same license is now used as a means to support other identity claims (such as age verification in order to purchase beer). Such usage is probably not particularly appealing to the original issuer of this assertion (the DMV probably really only wishes to make a statement about the holder's ability to drive, not their suitability to buy and drink beer!)

assertion of my authenticated status to its *service provider* (SP) partners in a *circle of trust*. Identity providers may also make other identity claims about someone, based upon information that a person has given them in association with that person's account held at the IdP. For example, it's possible for an identity provider to claim (on my behalf) that I am older than twenty-one, or that my email address is spam-me@no.spam.no. The user may control the behaviour of identity providers and service providers. For example, an SP may accept assertions issued by one of several identity providers, and the user of the SP can choose which identity to use. Or, the user can choose to be anonymous at the service provider.

## Making Identity Claims

"Do you think that when they asked George Washington for ID that he just whipped out a quarter?" – *Steven Wright, Comedian*

Here are some examples of identity claims –

i) spam-me@no-spam.no was authenticated via X.509 certificate holder-of-key at 2006-01-26T10:31:05Z is a claim of authentication status of a certain, named individual as of a certain time.

ii) John's birth month is May.

iii) Lois lives in Poughkeepsie, New York.

An identity provider may make the first of the above claims to a service provider. The service provider asks, "Is the user who just appeared at my site known to you, and authenticated?" This is an "authentication request", which may also result in sharing an identifier (the email address in this case) between the identity provider and the service provider. The Liberty ID-FF and SAML 2 specifications address exactly this concept – that an identity provider can make a claim of the authenticated status of a "security principal" (in many cases, simply a person) to service providers with whom the identity provider has some trusted relationship. In the Liberty specifications, the trust relationship between the service provider and the identity provider regarding these types of assertions is bi-directional – the identity provider wishes to know that the party to whom it is making an assertion is a party that it can trust, and vice-versa. In a case where the identity provider might be in some way (legally) liable for making this claim, this bi-directional trust is important!

As noted earlier, an identity claim might be made directly by me, or on my behalf, by an identity provider. Examples of each of these cases are shown below:

```
1. A personal profile query

  HTTP 200
  Content-Type: application/vnd.paos+xml
  Content-Length: 1234

  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

    <S:Header xmlns:wsa="http://www.w3.org/2005/03/addressing">

      <wsa:MessageID>uuid:C8797D0D-9020-07FC-AF0A-5622C01F4A61</wsa:MessageID>
```

```
      <wsa:Action>urn:liberty:id-sis:pp:2003-08:Query</wsa:Action>

      <wsa:ReplyTo xml:id="ReplyTo123">
        <wsa:Address>http://example.com/soap/Horoscope</wsa:Address>
      </wsa:ReplyTo>

      <wsa:To>http://www.w3.org/2005/03/addressing/role/anonymous</wsa:To>

    </S:Header>

    <S:Body>
      <pp:Query xmlns:pp="urn:liberty:id-sis-pp:2003-08">
        <pp:QueryItem>
          <pp:Select>/pp:PP/pp:Demographics/pp:Birthday</pp:Select>
        </pp:QueryItem>
      </pp:Query>
    </S:Body>

  </S:Envelope>

2. Service responds to request

  POST /soap/Horoscope HTTP/1.1
  Host: example.com
  Accept: text/html; application/vnd.paos+xml
  PAOS: ver="urn:liberty:paos:2005-12", "urn:liberty:paos:2003-08";
"urn:liberty:id-sis-pp:2003-08"
  Content-Type: application/vnd.paos+xml
  Content-Length: 2345

  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

    <S:Header xmlns:wsa="http://www.w3.org/2005/03/addressing">

      <wsa:MessageID>uuid:ab342ed-635ffee-142311ab-bedff67</wsa:MessageID>
      <wsa:RelatesTo>uuid:C8797D0D-9020-07FC-AF0A-5622C01F4A61</wsa:RelatesTo>
      <wsa:Action> urn:liberty:id-sis:pp:2003-08:QueryResponse</wsa:Action>
      <wsa:To>http://example.com/soap/Horoscope</wsa:To>

    </S:Header>

    <S:Body>

      <pp:QueryResponse xmlns:pp="urn:liberty:id-sis-pp:2003-08">
        <pp:Data>
          <pp:Birthday>--05-09</pp:Birthday>
        </pp:Data>
      </pp:QueryResponse>

    </S:Body>

  </S:Envelope>
```

Figure 1. Example Personal Profile

The example above shows (**emboldened**) a claim that the person was born on the 9th of May. And, in this case, the identity claim is used to allow the service provider to customize a web page based on the identity claim (in this case, knowing the birth month, the website can return a personalized horoscope page.) Such a claim can be made directly to the service provider, and the SP will have very little reason not to trust the claim. To

guarantee at least the integrity of the claim (the notion that some other party did not interfere with it between the making of the claim by a person, and that claim arriving at the SP) the claim might be digitally signed using a cryptographic key agreed between the SP and the claimant (perhaps generated individually for each SP and then encrypted using the public key of the SP to prevent collusion between SPs based on a single signing key supplied by the claimant.)

Here's an identity claim regarding authentication status of an individual:

```
<lib:AuthnResponse xmlns:lib="..." xmlns:samlp="..."
  ResponseID="hhuuja1bc744hGJn5Q9A5yvEIgS"
  InResponseTo="Zon3WjJ2KL7j+bJu7MuIr4Pt2go5" MajorVersion="1" MinorVersion="2"
  consent="urn:liberty:consent:obtained" IssueInstant="2002-10-31T21:55:41Z">

  <samlp:Status>
    <samlp:StatusCode Value="samlp:Success"/>
  </samlp:Status>

  <lib:Assertion MajorVersion="1" MinorVersion="2"
    AssertionID="e06e5a28-bc80-4ba6-9ecb-712949db686e"
    Issuer="http://IdentityProvider.com" IssueInstant="2001-12-17T09:30:47Z"
    InResponseTo="4e7c3772-4fa4-4a0f-99e8-7d719ff6067c">

    <saml:Conditions NotBefore="2001-12-17T09:30:47Z" NotOnOrAfter="2001-12-
17T09:35:47Z">
      <saml:AudienceRestrictionCondition >
        <saml:Audience>http://ServiceProvider.com</saml:Audience>
      </saml:AudienceRestrictionCondition>
    </saml:Conditions>

    <lib:AuthenticationStatement AuthenticationInstant="2001-12-17T09:30:47Z"
      SessionIndex="3" ReauthenticateOnOrAfter="2001-12-17T11:30:47Z"
      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
      <lib:Subject>
        <saml:NameIdentifier NameQualifier="http://ServiceProvider.com"
          Format="urn:liberty:iff:nameid:federated">342ad3d8-93ee-4c68-be35-
cc9e7db39e2b</saml:NameIdentifier>

        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
            urn:oasis:names:tc:SAML:1.0:cm:bearer
          </saml:ConfirmationMethod>
        </saml:SubjectConfirmation>

        <lib:IDPProvidedNameIdentifier
NameQualifier="http://ServiceProvider.com"
          Format="urn:liberty:iff:nameid:federated">342ad3d8-93ee-4c68-be35-
cc9e7db39e2b</lib:IDPProvidedNameIdentifier>

      </lib:Subject>

    </lib:AuthenticationStatement>

    <ds:Signature xmlns:ds="...">...</ds:Signature>

  </lib:Assertion>

  <lib:ProviderID>http://IdentityProvider.com</lib:ProviderID>
  <lib:RelayState>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</lib:RelayState>

</lib:AuthnResponse>
```

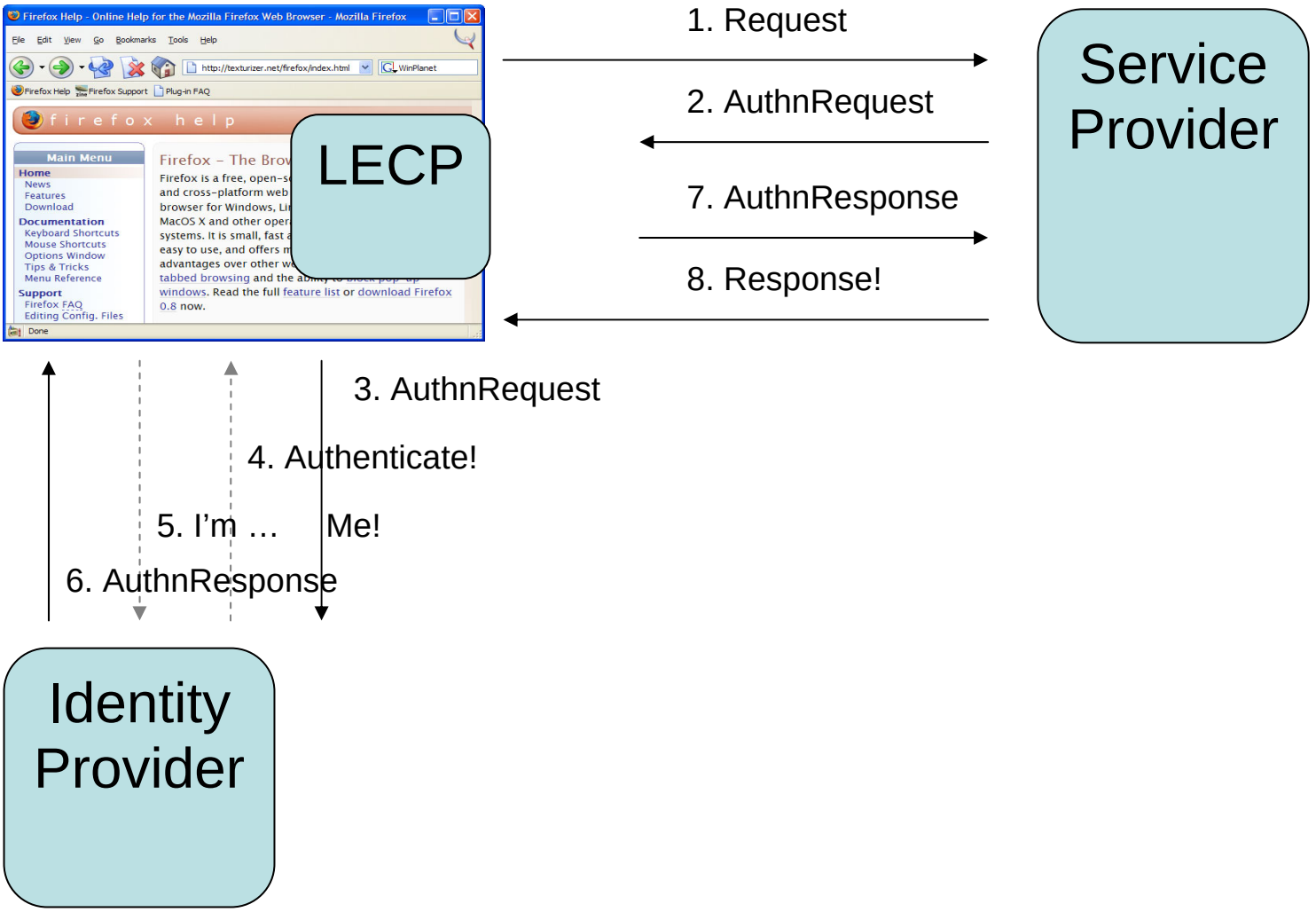
Figure 2. Example Authentication Assertion

In Figure 2, you'll see a much more complicated claim than in Figure 1. The complication arises from the fact that this is an identity provider making an assertion of a user's authentication status to a service provider. Below that claim (immediately below the emboldened section) is the digital signature of the identity provider, which can be used to verify that this claim was made by the holder of the key with which the assertion was signed.

This subtle but important difference is what drives the barkeep to better trust the information in your driving license (asserted by the DMV) rather than directly trusting the information verbally provided by yourself. It is all a matter of *trust*. Typically, the claims of authentication status such as in Figure 2 are made within the context of web browser-based single sign-on (SSO). In such a case, the service provider uses an HTTP redirect operation to redirect the user's web browser to the identity provider's website in order for the identity provider to check the user's authenticated status. In the existing world of the web, this is the only way that single sign-on can be achieved, but it can result in the user having little control over this essentially automatic process. However, the Liberty ID-FF specification offers another way, which can give the user of a piece of software such as (but not limited to) a web browser more control over the provision of identity claims.

## *Liberty-Enabled Client or Proxy Profile*

As noted above, in many cases, a user's web browser will be redirected from a service provider's website to the identity provider's site, without any action by the user. This can result in a loss of control over SSO by the user. An alternative to this process is for the




web browser to advertise to a service provider that it can locate an appropriate identity provider, and act as an intermediary between the IdP and the SP. This is done using the Liberty-Enabled Client or Proxy (LECP) Profile, which is shown diagrammatically below.

Figure 3. LECP profile

In the profile shown above, the web browser user-agent makes its first request (step 1.), and advertises (via an HTTP header) that it can accept an authentication request, and knows how to find an appropriate identity provider to service the request. The SP responds with an authentication request, which is directed to an identity provider chosen by the LECP. In step 3. the authentication request is forwarded to an appropriate IdP – note that the LECP itself might choose to act as an identity provider if the service

provider will accept an authentication assertion made by the user-agent itself. At this point, the LECP can present its user with choices about which IdP to use (for example, they user may have multiple identities, or wish to act pseudonymously, and the LECP can present these choices to the user). In step 3. the authentication request is forwarded to the chosen identity provider, and in 4, 5 and 6, the usual authentication steps occur as necessary before the LECP then forwards the authentication response obtained to the service provider. In step 8, the web browser receives its response to the initial request.

### *Identity Services*

The pattern used in the LECP profile can be generalized to include the ability for a user-agent (a web browser for example) to provide all kinds of identity claims, under the direct control of the user[2] . The following diagram shows how a service provider can query a user-agent in order to provide customized content to that user, based on an identity claim being made.

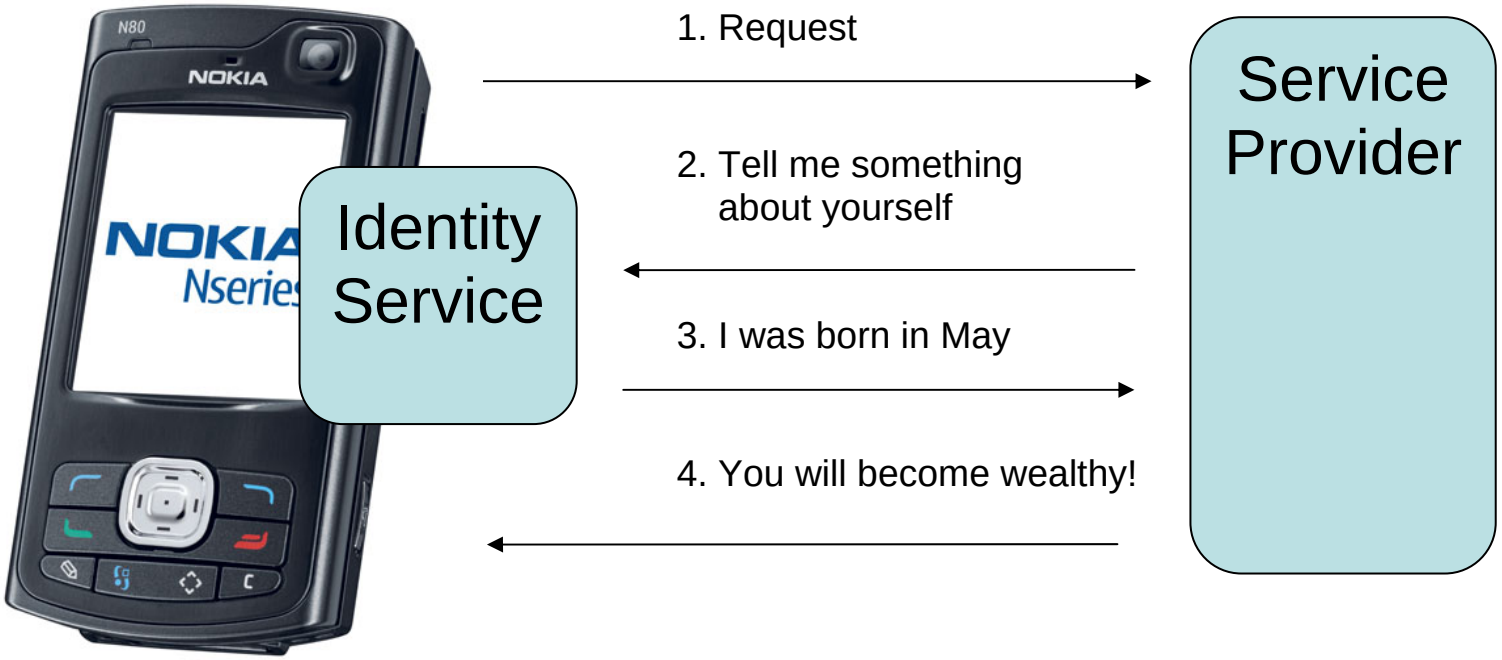



Figure 4. An identity service all of my own.

In the above example, software that can make identity claims is running directly on a person's mobile phone. Such software could implement any of the services defined by the Liberty ID-WSF specifications, including an Authentication Service, Single-sign-on Service, Discovery Service or Personal Profile Service. The example shows an individual

[2] Note the difference between self-asserted claims and trusted-third-party-asserted claims, as mentioned in previous sections,

using their mobile phone to access some service (perhaps via a web browser on the device). The service provider can then make a request for identity information directly to the service running on the device. This identity information could be digitally signed by the software application on behalf of the individual whose identity information is being exposed. In this particular case, the mobile phone is not directly addressable from the network and must thus always make the first request, and in doing so, advertise its service capabilities to the service provider. This advertisement is done using PAOS – the Reverse SOAP Binding of HTTP (see reading list below).

## Verifying Identity Claims

When an identity claim has been made, it may be important for a service provider to verify that claim, and to authenticate the presenter of the claim. Verification involves evaluating the evidence supporting the claim and determining whether that evidence is adequate. Such evidence may include an assertion stating essentially that "the presenter of this claim is who he says he is" – an authentication assertion. Thus, authentication of the presenter of the claim is a quite important piece of evidence supporting other claims made by that presenter. Of course, that authentication assertion may have been produced by someone other than the presenter of the assertion. For example, my identity provider claims to have authenticated me, and I present this assertion to a service provider. In which case, verification of my identity claim may involve i) verifying that I am who I say I am, which involves ii) verifying that the IdP who authenticated me is who he says he is and iii) ensuring that the identity claim is closely tied to the identity provider's claim of my authenticated status. To return to the driving license analogy, verification of my age depends on evidence that the person whose age is listed on the license is in fact me (perhaps the license has my photo on it) and that the issuer of the driving license is the real DMV (often by means of some stamp or seal).

One way to verify certain properties of a claim (such as integrity of the data in the claim, or the authenticated status of the presenter of the claim) is a digital signature. For example, an assertion of my age may be digitally signed by me. This links the cryptographic key used to sign the identity claim with that identity claim. An assertion stating my authentication status will most likely be signed by the issuer of the assertion, linking it inextricably to the IdP.

Both the SAML assertion (or some part thereof) of Figure 2 and the personal profile response (or some part thereof) of Figure 1 may be signed. Both the issuer and the presenter of the claim may sign different parts of the claim data in order to satisfy some of the properties described above. The XML Digital Signature specification may be used to sign XML content. The OASIS Web Services SOAP Message Security Core and Profiles and Liberty ID-WSF Security Mechanisms specifications give additional guidelines specifically regarding SOAP message signatures.

Any entity in possession of the necessary cryptographic tools may sign a piece of XML content. This means that an identity provider such as my bank may sign this XML content with an *asymmetric* key (one half of a *key pair*), meaning that the message signature can be validated with the other half of the key pair, by an entity that does not have the bank's (private) half of the key pair. This basic concept underlies the use of X.509 certificates as a means of issuing identity claims. The bank's certificate contains its public key, and can be given to those who rely on claims issued by the bank. Often, certificates are issued by *certificate authorities*, establishing an explicit chain of trust, documented in the certificates themselves.

It is of course the case that an individual may possess an X.509 certificate of her own, signed by some certificate authority. However, this is today unusual, and public-key based certificates have some drawbacks when used by individuals. It is the case that the public half of an asymmetric key pair is generally given to many relying parties. It is more important for a bank to provide entities relying on its assertion some concrete notion of the bank's identity, than for the bank to protect the privacy of its identity information (such as its public key). For an individual, however, protecting her privacy can be quite important, and issuing a public key to multiple entities might allow those entities to collude and track that individual's behaviour – the public key then becomes an identifying handle for the user. So, if an individual uses an asymmetric key to sign an identity claim, and offers the other half of that key pair to entities relying on identity claims, the individual may have her privacy compromised. In some cases that may be fine, and in others, it may not. In some cases, the value of the transaction might be such that the individual must compromise privacy in order to balance the needs of the relying party to establish a secure and reliable system for *all* of its users. The use of a single signature key by an individual to sign multiple pieces of content provided to multiple other entities offers the chance for these entities to collude to determine the identity of the individual. Such a possibility may or may not be a real risk. Conversely, if a claim is not signed, it does not provide a very high guarantee to the recipient of the claim that the claim can truly be associated with the presenter of the claim. Fortunately, it is possible to use a *symmetric* key to sign an identity claim, and to present a different key to each relying party, so it is not always necessary for an individual to compromise his identity by using a public key. In such cases, however, the signing key must be transferred to the relying party in such a way as to obscure it from others (thus becoming public). This can be done using common encryption mechanisms.

We've talked so far only about the verification of claims made by an individual and his identity provider. But service providers must often make identity claims too. In many cases an identity provider will wish to limit its liability by only making assertions for a certain audience, and a service provider must give evidence to the identity provider that it is worthy of the identity provider's trust. Therefore, a service provider might well sign its authentication request, and/or include an identifier for itself that the identity provider may use to find out more information about the service provider before issuing it an assertion. In such a case, it is possible for the identity provider to know that it has issued assertions on behalf of the same user to multiple service providers (or multiple assertions on behalf of one user to the same service provider). This might allow the identity provider to track

a user's behaviour, which could be considered a privacy risk by an individual. That risk must be balanced against the risk of an identity provider making identity claims on my behalf to service providers to whom I do not wish to give that information to, simply because the identity provider cannot authenticate the service provider to whom it is giving the assertion. Furthermore, it is possible to mitigate the privacy risk by providing the ability for an individual to make identity claims herself via the LECP and identity service mechanisms described above, and facilities that allow a person to control the behaviour of any identity provider that issues claims on the person's behalf. While such identity claims may not provide the weight of evidence necessary for access to high-value services, it is certainly the case that they can be trusted for the same kinds of things for which they are trusted today in the offline world.

## Conclusions

"The case has, in some respects, been not entirely devoid of interest." -
*Sir Arthur Conan Doyle, (Sherlock Holmes) A Case of Identity, 1892*

Identity providing software makes claims about someone's digital identity. An identity provider feels comfortable making such claims based upon its relationship with the individual who provides it with the original identity information, and those to whom the IdP makes identity claims. A relying party feels comfortable accepting claims, based on information it knows about both the presenter of claims, and the issuer. It is possible for the owner of the digital identity to operate his own identity provider – where the issuer of the claim is also its presenter. And in several cases, self-asserted identity information may be perfectly adequate for a service provider to determine whether it should provide service to a particular individual.

The Liberty ID-FF and ID-WSF specifications, by means of the LECP profile, and by the possibility to host identity services on client systems (such as personal computers and mobile phones) allow individuals to maintain some direct control over the release of identity claims. Furthermore, Liberty identity providers and service providers are free to offer facilities that allow their users some control over network-hosted personal identity data.

The Liberty specifications make appropriate use of existing technologies for anchoring trust in a networked environment (such as X.509 certificates and XML Digital Signature) but do not demand their use. Thus, in environments where it can be expected that such technologies are not deployed (such as in user-operated devices), it is still possible to deploy implementations of the Liberty specifications that allow an individual to maintain more control over his digital identity information.

## Further Reading

"We are here and it is now. Further than that all human knowledge is moonshine." -
*H. L. Mencken (1880 - 1956)*

Liberty LECP profile - http://www.projectliberty.org/specs/draft-liberty-idff-bindings-profiles-1.2-errata-v2.0.pdf

Liberty PAOS - http://www.projectliberty.org/specs/liberty-paos-v1.1.pdf

Liberty Security & Privacy Overview - http://www.projectliberty.org/specs/liberty-idwsf-security-privacy-overview-v1.0.pdf

Liberty Client Profiles - http://www.projectliberty.org/specs/liberty-idwsf-client-profiles-v1.1.pdf

XML Digital Signature - http://www.w3.org/TR/xmldsig-core/

OASIS Web Services Security - http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf