Privacy Postures of Authentication Technologies

Francisco Corella, PhD
fcorella@pomcor.com

Karen Lewison, MD
kplewison@pomcor.com

Latest Revision: July 14, 2013*

Abstract

The development of a successful Identity Ecosystem requires an interdisciplinary approach, with close collaboration between professionals who specialize in the five dimensions of Identity: Law, Business, Policy, Technology, and Society. Unfortunately one of those dimensions, authentication technology, has become extremely complex and difficult to understand for experts in the other dimensions. In fact, several silos have emerged within authentication technology, and technologists immersed in one silo often find it difficult to understand the work being done in other silos. This is a serious obstacle to the development of an Identity Ecosystem. To overcome this obstacle, it is necessary to develop a conceptual framework that makes it possible to describe authentication technologies and their practical implications in terms that are both accurate and comprehensible to non-specialists, by abstracting away operational concepts from the technical details. As a step in that direction we survey and classify a wide range of technologies, and delineate their privacy postures.

1 Introduction

The development of a successful Identity Ecosystem requires an interdisciplinary approach, with close collaboration between professionals who specialize in the five dimensions of Identity: Law, Business, Policy, Technology, and Society. Unfortunately one of those dimensions, authentication technology, has become extremely complex and difficult to understand to experts in the other dimensions.

In fact, several silos have emerged within authentication technology, and technologists immersed in one silo often find it difficult to understand the work being done in other silos. For example, the privacy-enhancing cryptographic technologies used in U-Prove [3] and Idemix [4] pertain to a highly specialized field of mathematical cryptography. At meetings of the Internet Identity Workshop,[1] whose unusual unconference format facilitates free-flowing discussion, it was clear after the launch of the National Strategy for Trusted Identities in Cyberspace (NSTIC) that few computer scientists who did not specialize in privacy-enhancing

*This paper is a substantial revision of an earlier version, based on reader feedback. The feedback and the changes are described in a blog post [1] and comments on the post. The earlier version can be found in [2].

[1]The Internet Identity Workshop (IIW), focused on user-centric digital identity, is held twice a year in Mountain View, California. It has a web site at http://internetidentityworkshop.com.

cryptography could evaluate the practical benefits and drawbacks of U-Prove and Idemix and the practical differences between the two systems. Conversely, we conjecture that few mathematicians working on privacy-enhancing cryptography understand the complexities of OAuth and OpenID Connect. The documentation being developed by the OAuth working group [5] includes, as of this writing, four RFCs[2], eight Internet Drafts classified as working group documents, and thirteen Internet Drafts classified as related documents; and OpenID Connect is an extension of OAuth that adds seven specifications of its own [6] to the OAuth documentation.

This is a serious obstacle to the development of an Identity Ecosystem. To overcome this obstacle, it is necessary to develop a conceptual framework that makes it possible to describe authentication technologies and their practical implications in terms that are both accurate and comprehensible to non-specialists, by abstracting away operational concepts from the technical details. As a step in that direction we survey and classify a wide range of technologies, and delineate their privacy postures. Privacy is by no means the only aspect of authentication technology in need of clarification, but it is the most challenging, because insufficient attention has often been paid to the privacy implications of Internet technologies.[3]

The table in page 3 summarizes the privacy features of the authentication technologies that we will be discussing. Each row of the table refers to a technology or class of technologies. The subdivisions within the set of rows and the checkmarks in the first six columns sketch out a faceted classification of the technologies, which we find helpful in understanding what technologies provide what features. We discuss this classification below in Section 2. The other seven columns correspond to seven privacy features that can be used to roughly describe the privacy posture of each technique. We discuss the features in Section 3, and the privacy postures in Section 4, before concluding in Section 5.

2 Classification of Authentication Technologies

A key distinction to be made among authentication technologies is between those designed for two-party authentication, where a user authenticates to a service provider as a returning user, indicated by a checkmark in column 1 of the table, and those involving a third-party that identifies the user or asserts user attributes to a relying party, indicated by a checkmark in column 2.[4] Third-party authentication is called *federated authentication* when the third party and the relying party belong to different organizations.

Of course a third-party authentication technology can be used for returning user authentication by letting the third party and the relying party be one and the same. A checkmark in column 1 means that the technology can *only* be used for two-party authentication.

Another distinction that can be made among authentication technologies is whether the purpose of the technology is to identify the user (column 3), or to provide attributes that

[2]RFCs (Request for Comments) are formal publications of the Internet Engineering Task Force.

[3]Historically, little attention was paid to security and privacy in the early days of the Internet. A conscious effort was later made to improve security, and today most Internet-related specifications include a security considerations section. But privacy has lagged behind, in spite of great work done by early pioneers. More attention is being paid to privacy today, but few specifications still have a privacy considerations section.

[4]We use the generic term *service provider* to refer to an online service accessible through a web API, a web site, a web application, a back-end of a native application running on a mobile device, etc. A relying party is a service provider that learns user attributes from a third-party.

	1 Two-party authentication	2 Authentication by third party	3 Assertion of user identity	4 Assertion of user attributes	5 Closed-loop authentication	6 Open-loop authentication	7 Unobservability by identity or attribute provider	8 Free choice of identity or attribute provider	9 Anonymity	10 Selective disclosure	11 Issue-show unlinkability	12 Multishow unlinkability by different parties	13 Multishow unlinkability by same party
1. User ID & password	√		√		√		N/A	N/A	√	N/A		N/A	
2. User ID & generated OTP	√		√		√		N/A	N/A	√	N/A		N/A	
3. User ID & sent OTP	√		√		√		N/A	N/A		N/A		N/A	
4. Email address & P/OTP	√		√		√		N/A	N/A		N/A		N/A	
5. Microsoft Passport		√	√	√	√					√			
6. SAML browser SSO profile		√	√	√	√			(1)		(3)			
7. Shibboleth		√	√	√	√			(1)	√	(3)		√	√
8. OpenID (without PPID)		√	√	√	√			√		√			
9. ICAM OpenID profile		√	√	√	√			(2)	√	√		√	
10. OAuth		√	√	√	√			(2)		(3)			
11. OpenID Connect	√	√	√	√	√			√		(3)			
12. Uncertified key pair	√		√		√		N/A	N/A	√	N/A		N/A	
13. Public key certificate		√	√	√		√	√	N/A	√				
14. Structured certificate		√	√	√		√	√	N/A	√	√			
15. Idemix pseudonym	√		√		√		N/A	N/A	√	N/A		N/A	
16. Idemix anon. credential		√		√		√	√	N/A	√	√	√	√	√
17. U-Prove token		√		√		√	√	N/A	√	√	√		
18. ICAM BAE		√		√	N/A	N/A							

(1) User may choose provider from list presented by fourth-party service.
(2) User may choose provider from list presented by relying-party.
(3) Attributes selected by attribute provider or relying party.

may or may not uniquely identify the user (column 4), or both. A third party that uniquely identifies the user is called an *identity provider* while a third party that asserts user attributes is called an *attribute provider*.

Yet another distinction, indicated by columns 5 and 6, is between *closed-loop* and *open-loop* authentication, a terminology that we introduce in this paper. We refer to a party that issues or registers a credential as a *credential authority* (or a *certificate authority* when the credential is a certificate).[5] In third-party authentication the identity and/or attribute provider (IAP) is a credential authority. In two-party authentication the service provider is a credential authority. We say that authentication is *closed-loop* when the credential authority that issues or registers a credential is later responsible for verifying ownership of the credential at authentication time. We say that authentication is *open-loop* when credential ownership is verified instead by a relying party.[6]

Two-party authentication can only be closed-loop, but third-party authentication may be closed-loop or open-loop. Examples of third-party closed-loop authentication technologies include OpenID and Shibboleth. Examples of third-party open-loop authentication technologies include U-Prove, Idemix, and public key certificates.

Third-party closed-loop authentication gives rise to a privacy drawback that is peculiar to it. Since the IAP is responsible for verifying credential possession, it must be involved in the authentication event, and as a result it is informed of the fact that the user is using the credential for authenticating to a relying party, and of the identity of the relying party. The IAP can thus observe and record all the authentication events that make use of the credential, without having to collude with the relying parties to that purpose. This drawback is difficult to avoid.[7] We refer to it in column 7 as lack of *unobservability*.

The table highlights yet another classification facet by subdividing the rows of the table into four groups, according to how the user's identity or attributes are communicated to the service provider. This is useful because it groups technologies by technical similarity. The first group (rows 1–4) comprises technologies where the user authenticates to a service provider with a bearer credential.[8] The second group (rows 5–11) comprises technologies where a relying party receives a secondary bearer credential from an IAP via a web browser. The third group (rows 12–17) comprises technologies where the user's computing device proves possession of a cryptographic credential directly to a relying party. The fourth group (row 18) consists of a single technology whereby a relying party obtains user attributes directly from an attribute provider after identifying the user.

[5]Credentials such as certificates, U-Prove tokens or Idemix anonymous credentials are jointly created by the user's computing device and a credential authority. We say that those credentials are *issued* by the credential authority. Other credentials, such as passwords or uncertified key pairs, are entirely created by the user or the user's computing device, and then *registered* with the credential authority.

[6]We use the term "closed-loop" because the credential "loops back" to the issuer; "open-loop" simply means that there is no loopback. Also, the party that issues or registers the credential is "in the loop" in closed-loop authentication, but "out of the loop" in open-loop authentication.

[7]We proposed to avoid it by letting the browser hide the identity of the relying party from the IAP [7]; but that requires substantial modifications of standard browser functionality. More recently, Anil John has proposed hiding the identity of the relying party behind a proxy [8]; but then the proxy is able to observe the authentication events.

[8]A bearer credential is a datum that authenticates any party that produces it, without that party having to provide other evidence (such as knowledge of a private key).

3 Privacy Features

Columns 7–13 correspond to seven privacy features, and indicate which features are provided by each of the technologies.

Column 7, *Unobservability by identity or attribute provider*, is concerned with whether a third party that issues or registers credentials is able to observe how those credentials are presented by the user to relying parties. The unobservability feature is not applicable to two-party authentication.

Column 8, *Free choice of identity or attribute provider*, is relevant to third-party authentication technologies that lack unobservability. Since the credential authority observes how the credential is used, it matters to the privacy posture whether the user is free to choose a third party that she trusts. This issue is further discussed below in Section 4.2.

Column 9, *Anonymity*, shows whether or not a technology may allow the user to remain anonymous while authenticating to a service provider, in the sense of not having to provide personally identifiable information (PII) that uniquely identifies the user in a broader context than that of the service provider.

Column 10, *Selective Disclosure*, is relevant to third-party authentication, and refers to the ability to present to a relying party partial information extracted or derived from a credential. A simple form of selective disclosure is to be able to disclose some but not all of the attributes in a credential. A sophisticated example of selective disclosure, supported by Idemix, is to be able to prove that the user is old enough to purchase liquor based on a date of birth included in a credential without disclosing the date of birth.

Columns 11–13 are concerned with unlinkability, i.e. with the extent to which a credential can be used to track a user. The terminology comes from the literature on privacy-enhancing cryptography. *Issue-Show Unlinkability* (column 11) means that it it not possible to determine whether the same credential is involved in a given issuance event and a given authentication event. *Multishow Unlinkability by Different Parties* (column 12) means that it is not possible to determine whether the same credential has been used to authenticate to two different parties. *Multishow Unlinkability by Same Party* (column 13) means that it is not possible to determine whether the same credential has been used to authenticate to the same party on two different occasions.

There are two levels of multishow unlinkability. In *strong multishow unlinkability*, the relying party or parties cannot tell whether the same credential was used in two different occasions even if the credential issuer helps them by sharing the credential issuance information it has. In *weak multishow unlinkability* the relying party or parties cannot tell provided that the credential issuer keeps the credential issuance information secret. The table does not make a distinction between strong and weak multishow unlinkability: checkmarks in rows 12 or 13 indicate that either form of multishow unlinkability is provided.

Unlinkability is usually a concern in scenarios involving an IAP and multiple relying parties. However there are two-party authentication scenarios where issue-show unlinkability and multishow unlinkability by the same party would be relevant and desirable for the sake of user privacy.

For example, suppose that a web site providing high quality medical information for patients offers paid subscriptions. A user who visits the site after buying a subscription must be recognized as a returning user. However the site may want to provide anonymous access, to give the user confidence that their medical concerns are not relayed to medical

insurance companies. That requires issue-show unlinkability, so that user visits cannot be linked to payment information that would identify the user. Multishow unlinkability would further increase confidence by preventing the linking of multiple visits which in combination could potentially help identify the user.

None of the two-party technologies in the table provides unlinkability. However a third-party technology can be used for two-party authentication; so a service provider could issue U-Prove tokens to its users to provide issue-show unlinkability, or Idemix anonymous credentials to provide both issue-show and multishow unlinkability.

4 Privacy Postures

4.1 Group 1: User Authenticates to Service Provider with Bearer Credential

This group comprises ordinary passwords and one-time passwords (OTPs) used in conjunction with a user ID chosen by the user or an email address.

Ordinary passwords have serious security vulnerabilities, but they have one important privacy feature: they provide anonymity for returning-user authentication if used in conjunction with a user ID freely chosen by the user (row 1).

One-time passwords (OTPs) are short-term passwords generated at regular intervals in a sequence determined by a secret seed [9]. Like ordinary passwords, they are used in conjunction with a user ID or an email address; they may also be used in conjunction with an ordinary password or a PIN for two-factor authentication. OTPs can be generated by the user with a fob or an application (row 2), or sent to the user by text or voice messaging or by email (row 3). User-generated OTPs can provide anonymity if the user registers the OTP seed and a freely chosen user ID without providing any PII; we indicate this by an anonymity checkmark in row 2 even though this is not typically done today. Sending OTPs to the user voids anonymity because the service provider must be told where to send them.

Row 4 refers to an ordinary password or OTP used in conjunction with an email address, which voids anonymity.

Issue-show unlinkability and multishow unlinkability by the same party are not provided by passwords or OTPs, and other features besides anonymity are not applicable.

4.2 Group 2: Relying Party Receives Secondary Bearer Credential From IAP

This group includes seven third-party closed-loop authentication technologies (rows 5–11), which use similar browser-based mechanisms: the relying party directs the browser to the IAP, which recursively authenticates the user and directs the browser back to the relying party, sending a secondary credential to the relying party via the browser; any technology can be used for the nested authentication of the user to the IAP, but here we will only consider the usual case where the user authenticates to the IAP as a returning user with a user ID and a password or some other form of two-party authentication. The secondary credential is a bearer credential that discloses user attributes to the relying party and authenticates

the browser as acting on behalf of the user to whom the attributes apply.[9] The identity or attribute provider may authenticate the user directly upon presentation of a user credential (typically a user ID and a password, but sometimes a public key certificate and its associated private key [10]), or indirectly upon presentation by the browser of an authentication cookie demonstrating that the user has logged in earlier and a login session is in progress. Some of the seven technologies require the IAP to ask the user's consent before disclosing attributes to the relying party, some do not. When consent is not required, third-party authentication takes place without user intervention once the user has logged in to the IAP, and is therefore a form of *single sign-on* (SSO).

The following technologies are included in this group:

- The historical *Microsoft Passport* [11] (row 5), later called *Windows Live ID* and now simply referred to as Microsoft account. It was originally positioned as single sign-on for ecommerce, but had to be repositioned after being criticized on privacy grounds.

- *SAML Browser SSO Profile* [12] (row 6), an OASIS standard used for federated authentication by corporations and government organizations, and its derivative *Shibboleth* [13] (row 7) used by research and education institutions.

- *OpenID* [14], a standard of the OpenID Foundation (row 8), and the *Identity, Credential and Access Managment (ICAM) profile of OpenID 2.0* [15] for authentication of citizens conducting transactions with the Federal government at Level of Assurance (LOA) 1 (row 9).

- *OAuth* [5], an IETF proposed standard often used for *social login*, where a user authorizes a web site or application to access her account at a social network and to obtain her identity and attributes from the account (row 10).

- *OpenID Connect* [6] (row 11), a feature-rich standard under development by the OpenID Foundation, which extends OAuth with features found in OpenID, as well as other features [16], including a *Self-Issued OpenID Provider* feature for implementing two-party authentication (as indicated by the checkmark in row 11, column 1) and a feature that allows the relying party to fetch user attributes directly from an attribute provider, in a manner similar to the ICAM Back-end Attribute exchange discussed below.

As third-party closed loop authentication technologies, these technologies keep the identity provider informed of the user's authentication events, and thus lack unobservability.

Microsoft Passport allowed no choice of IAP: Microsoft was the only one. SAML Browser SSO profile provides a mechanism for allowing the user to discover available IAPs and Shibboleth uses it to implement a fourth-party service that allows the user to indicate the IAP of her home institution. OpenID provides full freedom of choice, allowing the user to choose an IAP unknown to the relying party. When it was introduced, OpenID evangelists emphasized that a developer with access to a web server could create her own IAP and have it accepted by any relying party. The ICAM OpenID profile restricts the choice of IAP to a list of ICAM-approved providers that the relying party presents to the user. OAuth requires the

[9]The attributes may be disclosed in a signed statement, or they may be retrievable by the relying party from the identity or attribute provider upon presentation of the secondary credential.

relying party to register with the IAP in order to establish a shared secret, so the user can only choose among those IAPs that the relying party has registered with. OpenID Connect is an extension of OAuth, but provides freedom of choice with features that allow the relying party to learn the user's preferred IAP and register with it on the fly.

The technologies in this group are primarily intended to identify the user, but two of them can be used to provide attributes while preserving anonymity. Shibboleth makes no distinction between user identity and user attributes, the user's identity being defined by one or more attributes, and the IAP is free to only disclose to the relying party attributes that do not uniquely identify the user. This is indicated by a checkmark in column 9. OpenID makes a distinction between a "Claimed Identifier", which is supported by the main OpenID 2.0 protocol, and user attributes, which are supported by protocol extensions. Theoretically, the identity provider may omit the claimed identifier and only include attributes in the secondary credential, thus providing anonymity. However, we doubt that OpenID has ever been used in this manner, so we have not included a checkmark in column 9. The ICAM profile of OpenID provides anonymity by requiring the identity provider to supply different identifiers for the same user to different relying parties [15, §3.4.2]. According to Nat Sakimura [16], such *Pairwise Pseudonymous Identifiers (PPID)* were used by Google and Yahoo before being required by the ICAM profile of OpenID. Row 8 refers to the traditional use of OpenID without pairwise pseudonymous identifiers.

In the context of third-party closed-loop authentication, selective disclosure refers to the selection of the attributes that the identity provider discloses to the relying party by means of the secondary credential. The selection is often made by the identity provider and/or the relying party without user input at authentication time. However, in Microsoft Passport, according to [11], the user was able to choose the attributes to be disclosed; and according to [17], OpenID identity providers allow the user to choose among optional attributes to be conveyed to the relying party using the OpenID Attribute Exchange extension [18] (not be confused with the ICAM Backend Attribute Exchange of row 18 discussed below) or the Simple Registration extension [19]. In OpenID (including the ICAM profile), OAuth, and OpendID Connect the user is asked for consent before the attributes are disclosed, while in the SAML Browser SSO profile she is not. In traditional implementations of Shibboleth the user is not asked for consent, but the *uApprove* extension [20] developed by SWITCH, the Swiss InCommon federation, does ask for consent. The Scalable Privacy NSTIC project [21] is developing a privacy manager that will let a Shibboleth user choose what attributes are disclosed.

Most of the technologies in the group lack unlinkability because they lack anonymity. Shibboleth provides weak multishow unlinkability by different parties and by the same party when it only discloses attributes that do not uniquely identify the user. The ICAM profile of OpenID provides weak multishow unlinkability by different relying parties.

4.3 Group 3: User's Device Proves Possession of Cryptographic Credential to Service Provider

The first three technologies in this group (rows 12–14) are all based on public key cryptosystems, but they have very different security postures:

- An *uncertified key pair* (row 12) pertaining to a public key cryptosystem that provides

digital signatures, such as RSA, DSA or ECDSA, can be used for two-party authentication to a service provider by registering the public key with the service provider and later demonstrating knowledge of the private key by using it to sign a challenge [22]. It has the same privacy posture as an ordinary password (but much stronger security!).

- A *public key certificate* (row 13) binds a public key to one or more attributes that may or may not uniquely identify the user. The user demonstrates knowledge of the associated private key when she presents the certificate. In a common set-up a certificate authority issues an X.509 certificate [23] which the browser presents as a TLS client certificate [24]; the certificate and associated private key may be stored in a smartcard (e.g. a PIV card) connected to the user's computer and accessible to the browser. But alternative certificate formats have emerged, such as the JSON Web Token used as one authentication option in Mozilla's BrowserID/Persona technology [25].

  A known flaw in the TLS protocol causes the client certificate to be sent in the clear, causing global observability. However this should be viewed as a security issue of SSL/TLS rather than a privacy flaw of public key certificates. Discounting this flaw, public key certificates provide unobservability by the issuer and may also provide anonymity, since the attributes in the certificate do not necessarily identify the user. However they do not provide selective disclosure or unlinkability.

- A *structured certificate* (row 14) [26] binds a public key to a hash tree of attributes, allowing the user to choose which attributes are hashed and which are disclosed when the certificate is presented. It has the same privacy posture as an X.509 public key certificate except that it provides selective disclosure, thus occupying a middle ground between public-key cryptography and privacy-enhancing cryptography.

The last three technologies in the group (rows 15–17) are concerned with Idemix and U-Prove, two privacy-enhancing authentication technologies that have been implemented and are being used in several pilots.

Idemix [4] features two kinds of credentails: *Idemix pseudonyms* (row 15) and *Idemix anonymous credentials* (row 16). An Idemix pseudonym is used for two-party returning-user authentication. It is similar to an uncertified key pair[10] and has the same privacy posture. An Idemix anonymous credential, on the other hand, is used for third-party open-loop authentication, and provides full privacy, including unobservability, anonymity, selective disclosure (including the ability to prove that a numeric attribute is greater or less than a numeric constant without disclosing its value), issue-show unlinkability, and strong multishow unlinkability by the same party or different parties.

U-Prove [3] (row 17) features *U-Prove tokens*, which are used for third-party open-loop authentication. They provide unobservability, anonymity, selective disclosure (but not the ability to prove that a numeric attribute is greater or less than a numeric constant without disclosing its value), and issue-show unlinkability. However they do not provide multishow unlinkability. To avoid being tracked by token presentations to relying parties, the user

[10] It differs from an uncertified key pair in that it embeds two secrets, a master secret shared with all of the user's pseudonyms and anonymous credentials, and an additional random secret specific to each pseudonym; and in that it can be used in conjunction with anonymous credentials in zero-knowledge proofs of knowledge.

must obtain multiple tokens from the issuer, and present different tokens to different relying parties. U-Prove provides an efficient parallel procedure for creating a batch of similar tokens at once, and refers to the batch as a *U-Prove credential*. U-Prove profiles have been implemented in a variety of technologies; the WS-Trust profile [27] was used in CardSpace.

4.4 Group 4: Relying Party Retrieves User Attributes from Attribute Provider

This group comprises a single technology, *Backend Attribute Exchange (BAE)* [28] (row 18), an ICAM protocol that allows a relying party to retrieve user attributes directly from an authoritative source after the user has authenticated and been uniquely identified. This protocol is exceptional: a credential is not used to obtain the attributes, so the technology cannot be classified as either closed-loop or open-loop authentication.

BAE has practical advantages, but not in the area of privacy. Since the attribute provider is queried by the relying party, it is informed of the authentication event; hence there is no unobservability; and the user does not choose the attribute provider. Since the user must be uniquely identified in order to retrieve her attributes there is no anonymity and no unlinkability. And the relying party can obtain any attributes it wants, without user consent, so there is no selective disclosure.

5 Conclusion

The complexity of authentication technology is a serious obstacle to the development of an Identity Ecosystem because it stands in the way of effective cross-disciplinary work. To overcome this obstacle we have called for the development of a conceptual framework that will make it possible to describe authentication technologies in terms that are both accurate and comprehensible to non-specialists, by abstracting away operational concepts from the technical details. As a step in that direction we have sketched a faceted classification of a wide range of authentication technologies, and evaluated those technologies against seven privacy criteria. We hope that this will stimulate further work towards facilitating an actionable understanding of authentication technologies.

References

[1] Francisco Corella. Feedback on the Paper on Privacy Postures of Authentication Technologies. May 15, 2013. /2013/05/15/feedback-on-the-paper-on-privacy-postures-of-authentication-technologies.

[2] Francisco Corella and Karen Lewison. Privacy Postures of Authentication Technologies. Version of May 1, 2013. http://pomcor.com/techreports/PrivacyPosturesMayFirstVersion.pdf.

[3] Christian Paquin. U-Prove Cryptographic Specification V1.1 Draft Revision 1, February 2011. There is no http URL for this document, but it can be downloaded by following links from http://www.microsoft.com/u-prove.

[4] IBM Research—Zurich. Specification of the Identity Mixer Cryptographic Library, Version 2.3.4, February 10, 2012. Downloadable from https://prime.inf.tu-dresden.de/idemix/.

[5] OAuth Working Group Documents. http://datatracker.ietf.org/wg/oauth/.

[6] OpenID Foundation. Welcome to OpenID Connect. http://openid.net/connect/.

[7] F. Corella and K. Lewison. NSTIC, Privacy and Social Login, May 2011. Position paper presented at the W3C Identity in the Browser workshop. http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_48.pdf.

[8] Anil John. Challenges in Operationalizing Privacy in Identity Federations - Part 2. October 8, 2012. http://info.idmanagement.gov/2012/10/challenges-in-operationalizing-privacy.html.

[9] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. HOTP: An HMAC-Based One-Time Password Algorithm, December 2005. http://tools.ietf.org/html/rfc4226.

[10] Dartmouth College PKI Lab. Using PKI Authentication with Shibboleth, May 2003. http://www.dartmouth.edu/~pkilab/pages/ShibbAuthwithPKI.html.

[11] Brad Chase. Microsoft Passport: Streamlining Commerce and Communication on the Web. October 1999. http://www.microsoft.com/en-us/news/features/1999/10-11passport.aspx.

[12] J. Hughes et al. Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[13] Shibboleth Consortium. Shibboleth Home Page. http://shibboleth.net/.

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

[15] Terry McBride, Dave Silver, Matt Tebo, Chris Louden, and John Bradley. Federal Identity, Credentialing, and Access Management OpenID 2.0 Profile. November 2009. http://www.idmanagement.gov/documents/ICAM_OpenID20Profile.pdf.

[16] Nat Sakimura. Email message to the ID Commons community mailing list, May 9, 2013. Retrievable from the mailing list archive at http://lists.idcommons.net/lists/arc/community.

[17] Markus Sabadello. Comment on the blog post Feedback on the Paper on Privacy Postures of Authentication Technologies. July 14, 2013. http://pomcor.com/2013/05/15/feedback-on-the-paper-on-privacy-postures-of-authentication-technologies/#comment-907.

[18] D. Hardt, J. Bufu, and J. Hoyt. OpenID Attribute Exchange 1.0, December 2007. http://openid.net/specs/openid-attribute-exchange-1_0.html.

[19] J. Hoyt, J. Daugherty, and D. Recordon. OpenID Simple Registration Extension 1.0, June 2006. http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[20] SWITCH. uApprove — User Consent Module for Shibboleth Identity Providers. http://www.switch.ch/aai/support/tools/uApprove.html.

[21] Internet2. Scalable Privacy: An NSTIC Pilot for the Identity Ecosystem. https://spaces.internet2.edu/display/scalepriv/Scalable+Privacy.

[22] NIST. Entity Authentication, February 1997. FIPS PUB 196, http://csrc.nist.gov/publications/fips/fips196/fips196.pdf.

[23] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and CRL Profile, May 2008. http://datatracker.ietf.org/doc/rfc5280/.

[24] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2, August 2008. http://tools.ietf.org/html/rfc5246.

[25] Mozilla. BrowserID Protocol Overview. https://developer.mozilla.org/en-US/docs/Persona/Protocol_Overview.

[26] Francisco Corella. Structured certificates and their applications to distributed systems. Presented at RSA 2000. See also US Patent 6,802,002, assigned to Hewlett Packard Co.

[27] Christian Paquin. U-Prove WS-Trust Profile V1.0.
http://research.microsoft.com/apps/pubs/default.aspx?id=166974.

[28] ICAM. Backend Attribute Exchange (BAE) v2.0 Overview, January 2012.
http://www.idmanagement.gov/documents/BAE_v2_Overview_Document_Final_v1.0.0.pdf.