Pomcor's Response to the Notice of Inquiry On NSTIC Governance Structure

Francisco Corella, PhD
fcorella@pomcor.com

Karen Lewison, MD
kplewison@pomcor.com

July 22, 2011

The National Strategy for Trusted Identities in Cyberspace (NSTIC) has come at the right time. There has been much experimentation over the last decade with a variety of identity solutions for cyberspace. Unfortunately, the prevailing trend today is towards solutions that reduce privacy, security, user choice, and free competition. We trust that NSTIC will be able to reverse this trend, but it may not be easy.

Recognizing this difficulty suggests answers to two of the questions in the Notice of Inquiry (NOI):

2.2. While the steering group will ultimately be private sector-led regardless of how it is established, to what extent does government leadership of the group's initial phase increase or decrease the likelihood of the Strategy's success?

2.3. How can the government be most effective in accelerating the development and ultimate success of the Identity Ecosystem?

These are the questions that we want to address in this response to the NOI. We give our answers to those questions in Section 2 after discussing the state of the art in the next section.

1 State of the Art

1.1 Traditional Public Key Cryptography

A more secure alternative to password authentication has existed since the early days of the Web, when Netscape introduced the Secure Sockets Layer protocol (SSL); SSL was later taken over by the IETF and renamed Transport Layer Security (TLS). TLS allows the browser to authenticate by presenting a PKI certificate (as a TLS client certificate) and demonstrating knowledge of the associated private key. This is more secure than password authentication because the private key stays in the browser (or in a smart card accessible to the browser), and because it is practically impossible to guess.

PKI certificates have been used successfully in a number of trust frameworks worldwide. In the US, CAC and PIV smart cards carry PKI certificates and their corresponding private keys. But PKI certificates have not caught on as a universal authentication method on the Web; in fact, certificate authorities (CAs) seem to have given up on offering client certificates.[1]

There are several reasons for that. Among others:

1. Obtaining a certificate is a cumbersome process. To illustrate this we describe the process of obtaining a certificate for authentication to the US Patent and Trademark Office in Section 4.4 of [1].

2. There is no viable business model for a commercial CA to issue client certificates, because they compete with passwords, which are free.

3. Last but not least, CA-issued PKI certificates impinge on user privacy. Even if they only identify the user by a pseudonym, they allow user activities to be tracked by relying parties.

There have been two interesting efforts to realize the benefits of public key cryptography without relying on a commercial CA:

1. In WebID [2] the user authenticates with a self-signed certificate that binds a public key owned by the user's browser to the URL of a structured document within a personal Web page owned by the user, a page that participates in a web of trust. That URL serves as the user's identifier. To authenticate the user, the browser uses the self-signed certificate as a TLS client certificate. To verify the self-signed certificate, the relying party retrieves the structured document, extracts a public key from it, and verifies that the public key in the document coincides with the public key in the certificate.

2. In BrowserID [3, 4], the browser associates a public key it owns to an email address owned by the user, and asks the email service provider to make the same association. To authenticate the user, the browser presents the association to the relying party through a Javascript API and demonstrates knowledge of the private key. The relying party verifies the association by querying the email service provider. (To avoid the online query, the association that the browser presents to the relying party can be placed in a JSON Web Token signed by the email service provider; the signed token is equivalent to a certificate.)

The commercial CA is replaced by the user's personal page in WebID, and by the email service provider in BrowserID.

In spite of their drawbacks, we believe that PKI certificates have an important role to play in NSTIC. The fact that obtaining a certificate is a cumbersome process is just due to the fact that no efforts have been made to improve the

[1] A quick survey of Web sites of CAs shows that they offer TLS server certificates and email certificates, but no TLS client certificates.

process. In Section 2.1 of [1] we propose using a TLS extension to automate it. Once PKI certificates can be issued automatically using universally available TLS software, they could be issued by a variety of identity providers.

And while some uses of PKI certificates impinge on privacy, other uses do not; we provide examples of the latter in [1]. An important example is the case of a certificate issued by a site upon user registration and submitted to the same site on subsequent logins. Such a certificate replaces a username and a password, providing greater security with no reduction in privacy.

1.2 Privacy-Enhanced Credentials

The last decade has seen the development of credentials that provide more privacy than PKI certificates. Two types of such privacy-enhanced credentials have reached the stage where open source software is available for issuing and presenting credentials, viz. IBM's Idemix anonymous credentials [5] and Microsoft's U-Prove Tokens [6, 7]. Both allow the user to disclose to the relying party only a subset of the attributes asserted by the credential. Both preclude linkage between issuance and use of a credential, unless the linkage is made possible by the values of attributes disclosed to the relying party. Idemix credentials go further by allowing the user to prove inequality relations without disclosing the values of attributes involved in the relations,[2] and by precluding linkage between multiple presentations of the same credential, again unless linkage is made possible by the values of disclosed attributes.

Although privacy-enhanced credentials have been successfully implemented, they have yet to be successfully deployed on the Web. The Idemix system exists only in the form of a software library. The U-Prove system was deployed within the Information Cards framework, but Information Cards have been discontinued [8].

In [1] we argue that the main obstacles to the deployment of privacy-enhanced credentials originate from lack of support for such credentials in core Web protocols; to facilitate deployment, we argue in particular that the TLS protocol should be extended to allow the browser to present privacy-enhanced credentials, just as it presents PKI certificates, to a relying party playing the role of TLS server. It should also be extended to execute issuance protocols in addition to presentation protocols.

1.3 Double-Redirection Protocols

The last decade has also seen the development and deployment of a family of identity protocols based on a double-redirection mechanism, including Microsoft Passport [9] (now Windows Live ID), SAML Web Browser SSO Profile [10], Shibboleth [11], OpenID [12] and OAuth [13]. In these protocols, the relying party redirects[3] the browser to the identity provider, who authenticates the user

[2] E.g. proving that age is greater than 21 without revealing date of birth.

[3] Two different techniques are used to implement redirection: HTTP redirection, or form submission by Javascript code.

and redirects the browser back to the relying party, passing along a token that the relying party uses to establish the user's identity.

The identity provider usually authenticates the user by asking for a username and password. Therefore double-redirection procotols do not eliminate passwords, but they alleviate the user's need to remember different passwords for different relying parties.

The details of these protocols vary: for example, the token may be signed by the identity provider, or may be unsigned but verified by a direct query from the relying party to the identity provider. However they all have the following features in common:

1. The browser plays a purely passive role. It is not aware that authentication is taking place via double redirection.

2. The identity provider asks the user to enter a password after a redirection, if the user is not already logged in to the provider.

3. The identity provider takes responsibility for telling the user who the relying party list is.

4. The token is used as a bearer token, i.e. the relying party believes that the token refers to the user by the mere fact that the browser is able to produce it. The browser does not demonstrate knowledge of a private key or other secret.

These features result in drawbacks for security, privacy and user choice.

1.3.1 Security Drawbacks

One security drawback comes from asking the user to enter a password immediately after a redirection. It is well known that this facilitates phishing attacks [9, 14]

Another security drawback comes from the use of a bearer token. An attacker who obtains the token may be able to impersonate the user and log in to the relying party. In particular, a man-in-the-middle attack between the browser and the relying party will allow the attacker to log in as the user. This attack can be prevented by requiring the use of TLS when the browser sends the token to the relying party, i.e. by requiring the callback endpoint, where the token is sent, to be a TLS endpoint. Without TLS protection, a man-in-the-middle attack is easy to set up, for example, in a public WiFi network [15]. Nevertheless, this TLS requirement is sometimes overlooked or intentionally omitted.

For example, OpenID [12] does not require or recommend TLS for the callback endpoint. The same was true for the OAuth 2.0 specification, which is currently under development. After we reported this security issue, the working group agreed to at least recommend TLS [16]. Facebook argued that the specification should not go further and require TLS [17]. Today the Facebook developer documentation [18] uses the *http* URL scheme rather than *https* in examples of URLs of callback endpoints, thus suggesting that TLS not be used.

1.3.2 Drawback for User Choice

Because the browser is passive, the relying party must find out from the user which identity provider to use and how to access it.[4] This has turned out to be a hard usability problem. Lack of a good solution to this problem hampers user choice.

In OpenID, the user types in a URL which the relying party uses to discover the identity provider endpoint to which the user should be redirected. But this does not seem to have been successful since, as we shall see, the next version of OpenID will no longer use this mechanism.

Another way for a relying party to learn the user's identity provider is to display the icons of a number of identity providers and ask the user to pick one of them. But the number of icons that can be displayed with a reasonably simple user interface is limited.[5] This in turn limits the range of identity providers that a user can practically use to those whose icons are featured by most relying parties.

OAuth compounds this problem by requiring the relying party to register with the identity party. Most relying parties will register with only a few identity providers. As more and more users have Facebook accounts, relying parties may register only with Facebook and offer Facebook login as the only choice for social login. Some already do that today.

The OpenID Foundation has announced that the next version of OpenID will be OpenID Connect [19]. OpenID Connect will be built on top of OAuth and, like OAuth, will require registration of the relying party with the identity provider. This will radically change OpenID from a user-centric identity solution to one where the user is given little, if any, choice.

1.3.3 Drawback for User Privacy

Since the identity provider tells the user who the relying party is, it must have that information. This means that the identity provider must be informed of every login of the user to a relying party.

There are cases where this does not raise privacy concerns, e.g., when a protocol such as SAML SSO [10] is used for federated authentication across companies working on a joint project, or Shibboleth [11] is used for federated authentication across academic institutions.

But when double redirection is used for authentication on the Web at large, informing the identity provider of every login to a relying party is an invasion of the user's privacy.

This issue first arose when Microsoft introduced Passport. Later OpenID tried to avoid the issue by letting the user choose any identity provider, and even set-up her own personal identity provider. As we have seen in the previous

[4] By contrast, when credentials are stored in the browser, the browser decides what credential to use, after asking the user, if necessary.

[5] Developers refer to the problem of displaying a large number of icons of identity providers as the Nascar problem, evoking the many logos displayed on race cars.

section, however, this experiment in user-centric identity will end when OpendID Connect replaces the current version of OpenID. The latest double-redirection technologies, OAuth and the forthcoming OpenID Connect, are raising the issue again.

1.4 Social Login

Social login is an identity solution, pioneered by Facebook Connect, where a relying party uses a social site as identity provider to authenticate a user, and is furthermore granted limited authorization to access the user's account at the site. Authentication is actually a by-product of authorization: the relying party obtains the user's identity by accessing the user's account and reading identity data from the account.

Social login is by far the most successful identity solution, and its popularity is increasing rapidly. One reason for this is the fact that the relying party obtains access to the user's social context. In particular, the relying party is able to issue udpates on behalf of the user, which are seen by the user's friends.

Social login is used by all the major social sites (Facebook, LinkedIn, Twitter, MySpace) and by major multi-purpose sites such as Google and Yahoo. Facebook has recently overtaken Google as the top choice for social login [20] .

In Section 4.8 of [1] we show how social login could be implemented using delegatable credentials. Today, however, it is implemented using OAuth. We have already discussed OAuth as a double-redirection protocol. OAuth is used for social login because it is an authorization protocol: the "Auth" portion of the name is short for Authorization rather than Authentication. As in other double-redirection protocols, the second redirection passes a token from the identity provider, in this case the social site, to relying party. This token is what grants to the relying party limited access to the user's account at the social site.

We have seen that, like other double-redirection protocols, OAuth reduces privacy. We have also seen that it reduces user choice by requiring registration of the relying party with the social site. The registration requirement furthermore inhibits competition because it is a barrier to entry for new social sites: a social site startup will not be able to convince relying parties to register in large enough numbers to offer a useful social login feature to its users.

Although this may not be of direct concern to NSTIC, we must also point out that the registration requirement gives unprecendented power to the dominant social site, i.e. Facebook, over the Web. It will soon be necessary for most Web sites to offer users the option to log in with Facebook, because not doing so would keep a substantial number of users from signing up. Once a Web site offers that option, many users, perhaps a majority of users, will use it. Facebook then will have the power to shut out instantaneously many or most of the site's users by revoking its registration. Barring government regulation, Facebook can do that at its sole discretion. And it has already shown that it will revoke registrations without warning, with good or bad reasons [21]. This power comes

in addition to the power that Facebook already has over its users [22]. For the first time, a form of centralized control seriously threatens the Web.

2 Responses to Questions 2.2 and 2.3

There is a trend in the industry today towards identity solutions based on double-redirection protocols such as OpenID and OAuth, which intrinsically reduce privacy; and among double-redirection protocols, towards OAuth and derivatives such as OpenID Connect, which, in addition to reducing privacy, also reduce competition and user choice by requiring registration of relying parties with identity providers. To be successful, NSTIC must reverse this trend. This dictates our responses to questions 2.2 and 2.3 of the Notice of Inquiry.

Question 2.2

While the steering group will ultimately be private sector-led regardless of how it is established, to what extent does government leadership of the group's initial phase increase or decrease the likelihood of the Strategy's success?

Response

To increase the likelihood of success of NSTIC, the government must lead the initial phase.

If the private sector leads, the current trend will continue. Industry associations such as OIX and Kantara were formed to bring together identity providers and relying parties that have implemented double-redirection technologies. As they participate in NSTIC, they will naturally favor those technologies. Industry leaders such as Facebook, Google, Yahoo, Twitter and LinkedIn have all invested in OAuth, and some of them have made strong commitments to OAuth and OAuth derivatives such as OpenID Connect. They are unlikely to consider other alternatives if they lead NSTIC.

The government should lead NSTIC until momentum has shifted away from double-redirection and OAuth, towards alternative technologies that are compatible with the goals of NSTIC.

Question 2.3

How can the government be most effective in accelerating the development and ultimate success of the Identity Ecosystem?

Response

To be most effective, the government should fund pilots and proofs of concept based on technologies that increase privacy and security and foster competition and user choice. Scarce resources should not be spent on pilots based on OpenID,

OAuth, OpenID Connect and other double-redirection technologies that are imcompatible with the Guiding Principles of NSTIC.

The government should also consider using funds from the Small Business Innovation Research (SBIR) programs of NIST and other agencies to fund applied research on means of overcoming the obstacles to the deployment of privacy-enhancement technologies.

Pilots and proofs of concept should start as soon as possible. Delay will allow social login technology based on OAuth to alleviate a user's need to remember many passwords. While this in itself is a good thing, it will come at a heavy cost in reduced privacy, security, and user choice; and by preempting an important benefit and selling point of NSTIC, it will make it more difficult for NSTIC to succeed.

References

[1] F. Corella and K. Lewison. A Proposed Architecture for the NSTIC Ecosystem, July 17, 2011. Available at http://pomcor.com/whitepapers/ProposedNSTICArchitecture.pdf.

[2] WebID Incubator Group. WebID - Universal Login and Identity for the Web. At http://webid.info/.

[3] Mozilla Labs. BrowserID, A Better Way Sign In. At https://browserid.org/.

[4] Michael Hanson et al. Verified Email Protocol. At https://wiki.mozilla.org/Labs/Identity/VerifiedEmailProtocol.

[5] Jan Camenisch et al. Specification of the Identity Mixer Cryptographic Library, Version 2.3.1, December 7, 2010. Available at http://www.zurich.ibm.com/~pbi/identityMixer_gettingStarted/ProtocolSpecification_2-3-2.pdf.

[6] Christian Paquin. U-Prove Technology Overview 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.

[7] 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.

[8] Microsoft Identity and Access Team. Beyond Windows CardSpace, February 15, 2011. Available at http://blogs.msdn.com/b/card/archive/2011/02/15/beyond-windows-cardspace.aspx.

[9] David P. Kormann and Aviel D. Rubin. Risks of the Passport Single Signon Protocol. *Computer Networks*, 33:51–58, 2000. Available at http://avirubin.com/passport.html.

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

[11] Tom Scavo and Scott Cantor. Shibboleth Architecture Technical Overview, Working Draft 02, June 2005. Available at http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.

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

[13] OAuth Working Group. Web Page of the OAuth Working Group of the IETF. At http://datatracker.ietf.org/wg/oauth/charter/.

[14] Ben Laurie. OpenID: Phishing Heaven, January 19, 2007. Available at http://www.links.org/?p=187.

[15] P. Moceri and T. Ruths. Cafe Cracks: Attacks on Unsecured Wireless Networks. At http://www1.cse.wustl.edu/~jain/cse571-07/cafecrack.htm.

[16] Eran Hammer-Lahav, Editor of the OAuth Specification. Message to the OAuth Working Group, available at http://www.ietf.org/mail-archive/web/oauth/current/msg05846.html.

[17] David Recordon. Message to the OAuth Working Group, available at http://www.ietf.org/mail-archive/web/oauth/current/msg05953.html.

[18] Facebook. Authentication. Retrieved on July 21, 2011 from http://developers.facebook.com/docs/authentication/.

[19] Don Thibeau. OpenIDs Second Act: OpenID Connect, May 20, 2011. At http://openid.net/2011/05/20/openids-second-act-openid-connect/.

[20] eMarketer. Facebook Becomes Top Choice for Social Sign-In, April 28, 2011. At http://www.emarketer.com/Article.aspx?R=1008364.

[21] Jason Kincaid. Facebook's Ban Bot Leaves Some Developers Baffled (And Angry), June 25, 2011. At http://techcrunch.com/2011/06/25/facebooks-ban-bot-leaves-some-developers-baffled-and-angry/.

[22] Tini Tran. Activist Michael Anti Furious He Lost Facebook Account–While Zuckerberg's Dog Has Own Page, March 8, 2011. At http://www.huffingtonpost.com/2011/03/08/michael-anti-facebook_n_832771.html.