User Authentication with Privacy and Security

Unfunded Proposal to the NSF Secure and Trustworthy Cyberspace (SaTC) Program

Francisco Corella, PhD
fcorella@pomcor.com

January 2012

Project Summary

Intellectual Merit Statement

Public key certificates for user authentication have been available on the Web since 1995 but have failed to be deployed outside of specific niches due to practical difficultes concerning the issuance, revocation and selection of certificates. This project will investigate methods of overcoming these deployment obstacles. Specific objectives include designing efficient protocols for issuing and renewing short term certificates, designing a method by which a relying party can request a set of certificates that the browser will present simultaneously using an extension of the Transport Layer Security (TLS) protocol, and demonstrating how a Web site can request a set of credentials from the browser during registration, issue its own certificate upon successful registration, and request a different set of credentials including its own certificate for subsequent logins. The new protocols and protocol extensions will be implemented by building upon existing open source software.

Broader Impacts Statement

Password reuse and other password security flaws are major contributors to the lack of security in cyberspace. The project will make it possible and practical to use certificates instead of passwords for user authentication to Web sites and Web applications, thus making a major contribution to cybersecurity. The project will also increase privacy by allowing third party identity and attribute providers to vouch for the user without being informed of the transactions that user engages in. Finally, the project will set the stage for successful deployment of credentials based on privacy-enhancing technologies in the future. To promote industry adoption the results of the project will be actively and broadly disseminated. New protocols will be submitted to the World-Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) for standardization and all reusable code developed during the project will be made available as free and open source software. Most of the new software will be developed by student interns that will be trained and supervised by the Principal Investigator.

1 Motivation

One of the reasons for the lack of security in cyberspace is the use of passwords for user authentication. Mitigating the security risks of password use is the main purpose of the National Strategy for Trusted Identities in Cyberspace (NSTIC) [1].

Passwords are insecure for many reasons: they are easy to guess; they are hard to remember, and procedures used to reset a forgotten password are vulnerable; large numbers of passwords can succumb at once to a dictionary attack if a password database is breached, the dictionary attack being particularly easy if the passwords are hashed but not salted [2], and unnecessary if they are stored in the clear [3]. But the main reason why passwords are insecure is password reuse [4].

Password reuse is attributed to the fact that users have to remember passwords for more and more Web sites. Third-party login solutions address this problem by allowing a Web site, playing the role of relying party, to outsource user authentication to another site that plays the role of identity provider, so that passwords have to be remembered for fewer sites. Double-redirection protocols such as OpenID [5], SAML Browser SSO [6], Shibboleth [7], OAuth [8], and the forthcoming OpenID Connect [9] redirect the user's browser from the relying party to the identity provider, which authenticates the user with username or email address and password, and redirects the browser back to the relying party.

However, third-party login as implemented today on the Web does not solve the password reuse problem. An attacker can still set up a decoy Web site that asks for a username or email address and a password. Sophisticated and security-conscious users will avoid using the high-value credentials that they use to log in to their identity provider when they access other sites; but many users will not. The attacker will therefore be able to capture the credentials of many users and use them to try to log in to popular identity providers. Furthermore third-party login decreases security because it facilitates phishing attacks by asking for a password after a redirection.

Instead of reducing the number of sites for which users must remember passwords, we propose to avoid passwords altogether by using instead public key certificates and other cryptographic credentials. Public key certificates have been available for user authentication on the Web as SSL client certificates (now TLS client certificates) since 1995, but have failed to be deployed except in niche environments; we believe, though, that this is due to specific deployment problems that can be overcome. We have actually devised two solutions to those problems: a solution that can be deployed in the short term but sacrifices some of the privacy features of cryptographic credentials, and a more ambitious solution that may take longer to deploy but takes full advantage of those privacy features.

We have proposed the former in response to the most recent NSF SBIR Phase I solicitation, and we are proposing the latter here in response to the *Secure and Trustworthy Cyberspace* solicitation.

2 Results from Prior NSF Support

Our work on Web authentication is derived from a 2010 NSF SBIR Phase I project, entitled *Collaborative Information Retrieval from Multiple Real-Time Sources.* (Award number: 1013594; PI: Francisco Corella; amount: $150,000; period of support: July 1 to December 31, 2010; the PI has not received any other NSF support.)

Although an SBIR Phase I project is only intended to investigate feasibility, we obtained substantial results:

1. We invented the concept of a search API descriptor, a machine readable description of a Web API of a search engine. The concept is derived from the OpenSearch standard [10], which is used by most browsers to run queries entered by the user in the browser's search bar (omnibar in the case of Chrome) on a user-selected engine. An OpenSearch description document specifies the format of a query URL. Our API descriptors specify how to send GET or POST requests to a Web API and how to interpret an XML or JSON data structure that encodes a page of results returned by the API, accomodating the many variations that can be found in existing APIs.

   We used API descriptors in our multisearch engine Noflail Search [11]. Originally Noflail Search provided Bing results obtained through a Web API [12], with advanced user interface features including zero-result analysis [13] and a search history that facilitates the browsing of the result sets of multiple queries simultaneously [14]; results of other search engines were displayed as natively rendered by those engines in pop-up windows. API descriptors allowed us to use the Web APIs of search engines other than Bing without ad-hoc programming, and to display results from those engines within our own user interface, with zero-result analysis and a cross-engine search history that allows the browsing of result sets from different engines simultaneously. We plan to publish a formal specification of API descriptors in the future.

2. We invented a method of browsing realtime result sets, ranked by a combination of recency and relevance criteria, that are changing as they are being browsed. Result sets are paginated in the traditional way, but new results are shown with a brighter background, and pages having new results have brighter buttons in the page menu. We filed a patent application for

the invention and incorporated it into Noflail Search. (The visual effects can best be observed by entering a query on a hot topic and choosing the realtime search engine Topsy in the search engine panel.)

3. We identified flaws in the OAuth protocol, including a security hole that was eventually acknowledged and acted upon by the OAuth Working Group of the IETF [15], and we conceived an alternative protocol, PKAuth [16, 17], that corrected those flaws.

PKAuth was the beginning of our current work on Web authentication and authorization. The project required a protocol for accessing proprietary data sources for collaborative information retrieval, including search and collection of target documents. OAuth was a candidate protocol, but we found it impractical because it required prior registration of the search engine with every proprietary data source. The purpose of registration in OAuth is to establish a shared secret between an OAuth server, which provides access to a resource owned by the user, and an OAuth client, which accesses the resource on behalf of the user. In PKAuth we avoid the need for registration by using public key cryptography instead of symmetric cryptography, with authentication of the parties by TLS certificates.

We soon realized that the need for prior registration in OAuth was not just impractical for our purposes; it was also an ominous threat to privacy and user choice on the Web.

OAuth is used by Facebook and other social networks for *social login*, a form of third party login where the social network plays the role of identity provider (the term was coined by Janrain [18]); the relying party is granted limited access to the user's account at the social network and can thus identify the user and obtain identity-related data; it can also issue updates on behalf of the user, which is an important benefit for some relying parties. The registration requirement means that the relying party can only use as identity providers those social networks that it knows of and has gone to the trouble of registering with. More and more relying parties are giving the user only one choice: the dominant social network, Facebook. This may lead to a situation where most users must have a Facebook account just to be able to log in to other Web sites, and where Facebook is informed of most Web users' activities, whether or not those users wish to have a relationship with Facebook.

PKAuth is a possible answer to this threat because it does not require registration of the relying party with the identity provider and therefore it does not restrict the user's choice of identity providers to those supported by the relying party.

PKAuth provides more privacy than OAuth because it allows the user to choose a trusted identity provider of her choice; but it remains that the identity

provider is informed of the user's logins. The launch of the National Strategy for Trusted Identities in Cyberspace [19] made us realize that it is desirable to provide greater privacy by using cryptographic credentials, including credentials based on privacy-enhancing technologies [20].

This led us to investigate several approaches to the use of cryptographic credentials for Web authencitation. We engaged with the Internet identity community through blog posts and white papers (which can be found at the Pomcor Web site [21]), email discussions, and participation in workshops. We wrote a response to the NSTIC Notice of Inquiry [22, 23] and to the call for comments on the Cybersecurity Green Paper [24, 25].

Our work on Web authentication was well received. We were invited to show a poster on PKAuth at the 2011 *IDtrust workshop* held at NIST [26, 16] and to make a presentation and participate in a panel at the *Identity in the Browser* workshop organized by the World-Wide Web Consortium [27, 28]. We received constructive feedback at sessions called by us and others at two consecutive Internet Identity workshops, IIW 12 [29, 30] and IIW 13 [31, 32]. Jeremy Grant, head of the NSTIC National Program Office, called a session at IIW 13 to discuss how different technologies align with NSTIC guiding principles [33], which he said was motivated by our response to the NSTIC Notice of Inquiry. We were asked to participate in a panel at the NIST *Meeting on Privacy-Enhancing Cryptography* [34] and to launch the panel discussion by presenting slides [35] summarizing a series of blog posts that we wrote earlier on the prospects for using privacy-enhancing technologies in the NSTIC Identity Ecosystem [36].

Unfortunately, we did not have enough resources to work on Web authentication and collaborative information retrieval simultaneously. After the launch of NSTIC we judged that the work on Web authentication was more timely. Consequently we made the difficult decision of postponing further work on collaborative information retrieval and we did not apply for Phase II of the SBIR project. However this proposal (as well as the above-mentioned proposal of a shorter-term alternative solution) can be viewed as a result and follow-up of the work on Web authentication that we started during the Phase I project.

3 Technical Discussion of the Proposed Approach

3.1 Obstacles to the Deployment of Public Key Certificates

Public key certificates have been available for user authentication on the World Wide Web since Netscape introduced SSL client certificates in 1995, but they have seen little use outside of particular niches.

We believe that the lack of deployment of client certificates is due to three specific technical obstacles:

1. *Issuance.* No method has yet been devised for automatically issuing a certificate to a user and importing it into the user's browser (after successful proofing).

2. *Revocation.* The traditional method of revoking certificates using certificate revocation lists (CRLs) is only suitable for a closed environment such as an enterprise intranet or a federal agency where there is a single certificate issuer. It is not suitable for an open environment such as the Web, where a relying party would have to store and incrementally update the CRLs issued by an unlimited number of certificate issuers.

3. *Selection.* If multiple certificates are available in the browser, the browser is not well equipped for choosing which one to use. A heuristic that the browser can use is to present a certificate that has been issued by the site being visited, but the site may not have issued any of the certificates in the browser.

These obstacles have not been overcome earlier due to a lack of incentives to find alternatives to passwords, and to a disincentive to collaborate in the development of a common solution within a fiercely competitive browser industry.

But things are changing:

- Government initiatives such as the Comprehensive National Cybersecurity Initiative [37] and NSTIC show that awareness of cybersecurity threats is rising. The present solicitation bears witness to such increased awareness. Password insecurity is a major cybersecurity threat both as an attack vector and as an attack amplifier when a password database falls to some other vulnerability.

- The browser industry has matured since the days when it consisted of Netscape and Internet Explorer engaged in fierce competition. It now counts five major browsers that seem to understand the need for common standards. The World Wide Web consortium (W3C) has shown that it can bring together all the major browser manufacturers to develop standards such as HTML5. It has already engaged them in identity work through the Identity-in-the-Browser workshop and the public identity mailing list that was set up after the workshop [38]. Harry Halpin has stated the intention of the W3C to organize a workshop on the use of public key certificates on the Web [39, 40].

We believe that it is not utopic today to try to overcome the historical obstacles to the deployment of public key certificates for user authentication on the Web.

3.2 Two Solutions to the Credential Deployment Problems

We have devised two solutions to the problems that have hindered the deployment of public key certificates and, more generally, cryptographic credentials. The first solution, which we shall call the *redirect-to-issuer* solution, can be deployed in the short term; we have proposed it in response to the most recent NSF SBIR Phase I solicitation. The second solution, which we shall call the *credential-request* solution, is the one that we are proposing here in response to the *Secure and Trustworthy Cyberspace* solicitation.

In both solutions the issuance obstacle is solved by extending the functionality of the *keygen* element of HTML5 [41] so that, in response to a form submission that contains the element, the server downloads a certificate that the browser imports automatically. In this proposal we assume that the extended functionality will have been provided as part of the redirect-to-issuer solution implemented with SBIR funds or our own funds.

The redirect-to-issuer solution combines authentication by a TLS client certificate with double redirection. It is suitable for certificates that provide identity data such as that which a relying party can obtain today from an identity provider using OpenID or OAuth. The relying party uses PKAuth to redirect the browser to the certificate issuer, which plays the role of identity provider but uses the certificate rather than a password to authenticate the user before redirecting the browser back to the relying party. Since the certificate is presented to the Web site that issued it, the browser can use the above-mentioned heuristic to select the certificate; this solves the selection obstacle. And since the certificate is verified by the same party that issued it, there is no need to use a CRL; this solves the revocation obstacle. The user chooses the identity provider from a menu presented by the relying party but populated by the browser based on the certificates present in the browser.

The credential-request solution is more ambitious; it is suitable for any kind of cryptographic credential, and it allows for the presentation of multiple credentials simultaneously. There is no double redirection: the credentials are presented directly from the browser to the relying party. The selection problem is solved by having the relying party explicitly request the credentials it needs. The revocation problem is solved by using renewable short-term credentials. Our vision for the credential-request solution is described in more detail the next section.

The credential-request solution affords the user more privacy. The credential issuer is not informed of how the credentials it issues are used. Furthermore, privacy-enhancing credentials can be used instead of, or in addition to, public key certificates. But the short term redirect-to-issuer solution is simpler for relying parties and provides immediate revocation; hence it will not become obsolete

once the longer term credential-request solution is widely deployed. The two solutions can coexist, providing a trade-off between flexibility and privacy on one hand and simplicity and immediate revocation on the other. (An intriguing idea, which we may investigate in the future, is the possibility of combining double redirection with privacy-enhancing credentials to address the difficult revocation problem presented by those credentials.)

3.3 The Credential-Request Solution

In the credential-request solution, which we are proposing here, the relying party requests a set of heterogeneous credentials that are presented directly by the browser to the relying party.

3.3.1 Using Multiple Heterogeneous Credentials

We envision the use of a variety of credentials. (In this project, however, we will limit our implementations to public key certificates, leaving other kinds of credentials for future work.) All credentials are kept in the browser: requiring a secondary user agent for keeping credentials would be an obstacle to widespread adoption, as demonstrated by the failure of Windows Cardspace [42]. Possible kinds of credentials include:

- A public key certificate in X.509 format as profiled by the IETF [43], binding a public key to data such as:
  - Self-asserted personal data as currently provided by some OpenID providers [44], or
  - An email address certified by the email service provider that provided the address, possibly accompanied by other identity data, or
  - Credit card data (embedded in a certificate so that purchases can be secured by the fact that the browser has to prove knowledge of the corresonding private key when the certificate is presented), or
  - A WebID URI (see Section 4 below).
- A public key certificate in a format other than the IETF X.509 profile.
- A public key certificate binding a public key of the user to an internal identifier of the user's account at the relying party, issued by the relying party itself when the user registers, and used to identify the user on subsequent logins. Such a certificate does not need to expire or be revoked because the relying party checks whether the account is still valid when the certificate is

presented. Also, such a certificate raises no privacy concerns because there is no third party involvement. We shall refer to such a certificate as an *account certificate*.

- A non-public-key certificate that binds an attribute that uniquely identifies a user, e.g. a social security number, to an attribute of the identified user, e.g. US citizenship, without involving a public key. Examples of such certificates include an X.509 attribute certificate as profiled by the IETF [45] and a SAML AttributeStatement assertion [46].

- A structured public key certificate that uses a hash tree to provide selective disclosure of attributes [47, 48].

- A cryptographic credential based on a privacy-enhancing technology such as a U-Prove [49, 50, 51] that provides selective disclosure of attributes and issue-show unlinkability.

- A cryptographic credential based on a privacy-enhancing technology such as a Idemix [52, 53] that provides selective disclosure of attributes, issue-show unlinkability, and multi-show unlinkability.

- A delegatable credential issued by a social network that the browser can use to delegate to the relying party limited access to the user's account at the social network, including the ability to issue updates on behalf of the user, thus implementing social login with higher security and privacy provided by cryptographic credentials. The delegatable credential could be a public key certificate or a delegatable anonymous credential [54].

Our intention is to specify an architecture that accomodates an open-ended variety of credentials, so that different Web communities and different use cases can use different kinds of credentials.

3.3.2 Renewing Credentials

The need for revocation of third-party credentials will be avoided by issuing renewable short-term credentials, renewing a credential being more efficient than issuing a credential from scratch.

A public key certificate can be renewed efficiently by using the same key pair for the new certificate. We envision a renewal protocol where the certificate issuer only needs to communicate to the browser the new issue date, the new expiration date, and the new signature, the browser being able to construct the certificate from that data. Furthermore, since the issuer and the browser have a pre-existing relationship, the protocol can use symmetric-key cryptography for confidentiality

and integrity protection, without requiring public key cryptography for authentication and key exchange. We estimate that, if the state of California issued driver's license certificates for its 40 million drivers, those certificates could be renewed twice a day by a single server, using a bandwidth of less that 8 megabits per second.

A credential update protocol has been implemented in Idemix [53, Section 6.1.4]. A renewal mechanism for a more modern system of anonymous credentials that uses a bilinear map has been proposed in [55] by Camenisch et al.

3.3.3 Requesting and Presenting a Set of Credentials

We envision the relying party requesting a set of credentials, e.g. a credit card certificate and an email address certificate when making a purchase; or a public key certificate issued by the relying party plus a credential conveying self-asserted identity data that the user maintains in an online personal data repository; or a government-issued certificate contained in a PIV [56] or CAC [57] card, accompanied by an X.509 attribute certificate [45] binding the identity demonstrated by the public key certificate to a security clearance level.

Each request for credentials will be a conjunction of requirements, each requirement being specified by a credential-request HTTP header, the headers being conveyed to the browser in an HTTP response. Each header may specify a credential class, e.g. a VISA credit card certificate issued by a VISA-certified issuing bank; or it may specify a disjunction of requirements, e.g. "either a VISA certificate or a MasterCard certificate". An Object Identifier (OID) [58] will identify each credential class and specify the trust requirements that must be satisfied by credentials of the class (e.g. the VISA root certificate that must back the certificate of a VISA issuing bank). For technologies that provide selective disclosure, a credential class requirement may specify the attributes that must be disclosed, each attribute being specified by an OID.

The browser may have multiple credentials belonging to a given credential class; for example, it may have multiple VISA certificates issued by different banks. The browser may then allow the user to set one as the default, and/or may ask the user to choose one at run time, and/or may associate different credentials with different user-defined personas (e.g. a business VISA card could be associated with a work persona and a personal VISA card with a private persona), a persona being associated with each browser window or each browser tab.

We envision a major extension of the TLS protocol [59, Section 7.4.1.4] that will allow the browser to present multiple credentials of various kinds to the relying party. Whereas currently the client certificate is sent in the clear during the handshake phase, we envision credentials being presented during a new

credential-presentation phase following the handshake, with confidentiality and integrity protection. (The client certificate is now sent before the key exchange because, in theory, it may contain a Diffie-Hellman public key used for the key exchange; but certificates with such keys are rarely, if ever, used.) We propose to use the PRF facility of TLS to generate joint random material for use during credential presentation protocols. This will facilitate the use of credentials based on the common random string model [60], instead of credentials based on the the Fiat-Shamir heuristic [61] (justified by the random oracle model [62]).

In response to a request for credentials from the relying party, the browser will establish a new TLS connection and present the required credentials during the credential-presentation phase.

TLS has a session-resumption feature that allows client and server to establish a new connection based on the same master secret as a previous connection. The two connections are then said to belong to the same session. This concept of session, however, must be kept distinct from the concept of a login session. When the relying party requests login credentials, the browser will establish a TLS connection, declining to resume any previous connection with the same relying party in order to present the credentials. This will create a new TLS session, which may or may not be resumed later depending on whether or not the TLS session parameters are still in the session caches of the browser and the relying party. When the TLS session cannot be resumed during the same login session, the browser need not present the credentials again, and, for performance reasons, it should not present them. Login session maintenance can be accomplished by session cookies in the usual manner. For greater security it can also be accomplished using cryptographic credentials, but we will leave a detailed specification of cryptographic session maintenance for future work.

4 Relationship to Other Work

The *Online Certificate Status Protocol* (OCSP) [63] allows a party that relies on a public key certificate to check with the issuer whether the certificate is still valid. Compared to our redirection-to-issuer approach, OCSP is conceptually simpler, but practically more complex for the relying party, which does not have to deal with the client certificate at all in the redirection-to-issuer approach. Compared to the credential renewal used in our credential-request approach, OCSP has the advantage that revocation takes effect immediately, but the disadvantage that the issuer has to be online and the certificate status check adds to the latency of the authentication process. OCSP could be used for public key certificates in the credential-request approach for cases where revocation must take effect immediately.

In *double redirection protocols* the user typically authenticates to the identity provider with a password, which creates a phishing vulnerability. However, the user can also authenticate with a public key certificate [64], similarly to our redirection-to-issuer approach. A distinguishing feature of our redirection-to-issuer approach, however, is that the user's certificate is presented to the certificate issuer, avoiding the need for a CRL.

BrowserID [65], proposed by Mozilla Labs, allows a browser to demonstrate knowledge of a private key whose corresponding public key is bound to an email address by the email service provider that provides the addresses. The relying party verifies the binding by querying the email service provider using the WebFinger protocol over TLS. This approach is equivalent to the use of an email certifcate in our credential-request approach, but it is more complicated, more ad-hoc, and arguably less secure because it relies on a Javascript API [66].

WebID [67], formerly known as FOAF+SSL, uses a WebID certificate containing a public key and a WebID URI pointing to an RDF data structure embedded in a Web page where the public key is published. The certificate, which can be self-signed, identifies a user as the owner of that page. The page contains links to the pages of other users and can thus be seen as node in the social graph of a distributed social network. As pointed out above, a WebID certificate is one kind of credential that can be used in our proposed credential-request approach.

Once a relying party has obtained data that identifies a user, it can supplement that data by obtaining attributes about the user directly from an attribute provider; this is called *attribute exchange*. There are currently two attribute exchanges initiatives, Backend Attribute Exchange [68] for supplementing the identity information contained in a PIV certificate, and the Open Attribute Exchange Network (OpenAXN) [69] for supplementing the identity data provided by an OpenID provider. Our credential-request solution provides a better way of asserting attributes by allowing the browser to present identity credentials and attribute credentials simultaneously. This avoids the latency impact of having to contact the attribute provider during the transaction, and the privacy impact of having to inform the attribute provider that the transaction is taking place. More importantly, our solution gives the user control over what attributes are disclosed to what relying parties, whereas in an attribute exchange solution any relying party is able to query the attributes of any user; for OpenAXN, this is an important privacy issue; for BAE, this is a security issue, because some of the attributes of some of the users may be confidential.

5 Objectives

The project will have the following objectives:

Objective 1. Design a protocol for the efficient renewal of short-term public key certificates; implement a proof of concept with working code.

Objective 2. Design a method by which a relying party requests the presentation of multiple cryptographic credentials and the browser presents the credentials simultaneously using a TLS extension; implement a proof of concept with working code for the special case where the credentials are public key certificates.

Objective 3. Demonstrate how the credential-request technique makes it possible for a Web site to request a set of credentials from the browser during registration, issue its own certificate upon successful registration, and request a different set of credentials including its own certificate for subsequent logins.

Objective 1 is a follow-up to the extension of the keygen functionality for issuing certificates that we have proposed as part of the redirect-to-issuer proposal.

Objective 2 is a stepping stone towards the successful deployment of other cryptographic credentials besides public key certificates, including attribute certificates, privacy-enhancing credentials, and delegatable credentials for social login. In a future project, we also intend to develop a method of session maintenance that will use public key cryptography instead of cookies.

After achieving the first two objectives, Objective 3 will be an important step towards eliminating the use of passwords on the Web without sacrificing privacy.

This proposal fits into a plausible three-stage technology roadmap for NSTIC:

1. The first stage would use the redirect-to-issuer approach to facilitate the deployment of public key certificates and eliminate the need for passwords, with only small changes to browsers and no changes to TLS, at the cost of sacrificing some of the privacy benefits of certificates by informing the certificate issuer of how its certificates are being used.

2. The second stage, enabled by the present proposal, would greatly increase privacy by not involving the certificate issuer in the certificate presentation, and by letting the relying party issue its own certificate during registration in order to authenticate subsequent logins without third party involvement.

3. The third stage would achieve full privacy through the deployment of credentials based on privacy-enhancing technologies, which would be presented during the TLS credential-presentation phase introduced in this proposal.

However the benefits of this proposal will only be realized if the industry adopts the techniques that will developed during the project. Promoting industry adoption is the goal of the Transitions phase of this proposal.

6 Work Plan

To achieve Objective 1 we will:

1. Informally specify an efficient protocol where a browser asks a certificate issuer to renew the certificate, and the browser responds by sending a new issuance date, a new expiration date, and a new signature, using symmetric keys for confidentiality and integrity protection.

2. Informally specify a further extension of keygen so that the certificate issuer and the browser agree on symmetric keys for the renewal protocol.

3. Implement client and server side libraries for the protocol and integrate the client side library into an extension of the open-source Firefox browser.

A formal specification of the protocol will be provided if the Transitions phase is funded.

To demonstrate that Objective 1 has been achieved we will implement a proof-of-concept of a certificate issuer using the server side library, issue a certificate to the extended Firefox browser, and renew the certificate. We will measure the bandwidth taken up by renewal of one certificate and the computational cost for the server, and verify by extrapolation that the above-mentioned estimates of the costs of renewing 40 million certificates twice a day are correct.

We estimate that achieving Objective 1 will take approximately 12 calendar months. The PI will be responsible for the informal specifications, the design of the software, and part of the programming. Most of the programming will be done by student interns trained and supervised by the PI.

To achieve Objective 2, we will:

1. Informally specify a major extension of the TLS protocol featuring a new credential-presentation phase, following the TLS phase, where the browser can present multiple credentials of different kinds.

2. Informally specify the presentation of a public-key certificate during the new TLS phase, with a proof of knowledge of the corresponding private key.

3. Extend the TLS library "Network Security Services" (NSS) [70] used by Firefox so that it can present multiple public key certificates to the server during the new TLS phase.

4. Extend the mod_ssl module [71] of the open-source Apache Web server, and the OpenSSL library [72] that it uses, so that they can accept multiple public key certificates presented during the new TLS phase, parse those

certificates, and make certificate data available to application code (e.g. written in PHP).

5. Informally specify HTTP headers that a relying party can use in an HTTP response message to request a set of cryptographic credentials as envisioned above in Section 3.3.3.

6. Extend Firefox so that it recognizes the extensions and presents the requested certificates.

Formal specifications will be provided if the Transitions phase is funded.

To demonstrate that Objective 2 has been achieved we will implement a proof-of-concept of a relying party that will request multiple certificates and will use the extended mod_ssl module to accept them.

We estimate that achieving Objective 2 will take approximately 15 calendar months. The PI will be responsible for the informal specifications, the design of the software, and part of the programming. Most of the programming will be done by student interns trained and supervised by the PI.

To achieve Objective 3 we will:

1. Implement a proof-of-concept Web server that will use the keygen extension to issue a renewable email address certificate, and will renew the certificate upon request by the user's browser. This server will simulate an email service provider.

2. Implement a proof-of-concept Web server that will use the keygen extension to issue a renewable personal data certificate, and will renew the certificate upon request by the user's browser. This server will simulate a personal data store [73], which a user may use to maintain self-asserted personal data in a single location for Web-wide use.

3. Implement a proof-of-concept Web server that will play the role of relying party and will feature:

   (a) A relational database containing a table of user accounts and a table of login sessions.

   (b) A user registration process, in which it will ask for an email address certificate and a personal data certificate in order to create a user account, and will issue an account certificate upon successful registration.

   (c) A user login process, in which it will ask for an account certificate and a personal data certificate, upon presentation of which it will update

the user's account with the personal data as needed, create a login session record, and set a session cookie in the browser.

(d) A user logout process that will delete the login session record and invalidate the cookie in the browser.

To demonstrate that Objective 3 has been achieved we will use the extended Firefox browser of Objective 1 to: request an email certificate and a personal data certificate from the respective servers; register with the relying party by presenting those certificates, and receive an account certificate; log in to the relying party and receive a session cookie; access the relying party using the session cookie for authentication; and log out from the relying party.

We estimate that achieving Objective 3 will take approximately 9 calendar months. The design of the software and some of the programming will be carried out by the PI. Most of the programming will be done by student interns trained and supervised by the PI.

The PI for this proposal is also the PI for the above-mentioned pending SBIR Phase I proposal. If the SBIR proposal is accepted, and if there is flexibility on the start date of this project, we would like to start working on this project immediately following the SBIR Phase I project. SBIR Phase I is scheduled in principle from July 1st to December 31st, 2012, but we expect to finish earlier than December 31st by starting at our own risk as soon as we get positive feedback from the cognizant program officer (expected in March). We will later plan the SBIR Phase II work so that the PI has ample time to work on both projects.

7 Conclusion

If successful, the project will overcome the obstacles that have so far hindered the deployment of public key certificates instead of passwords for user authentication on the World Wide Web, and will thus make a major contribution to cybersecurity. It will also increase privacy by allowing third parties to provide the user with credentials without being informed of how those credentials are used, and it will set the stage for the successful deployment of credentials based on privacy-enhancing technologies in the future; it will thus contribute to the security-with-privacy goals of the National Strategy for Trusted Identities in Cyberspace.

Transitions Phase

1 Introduction

The goal of the Transitions Phase will be to facilitate the adoption by the Web technology industry of the techniques developed during the project. To that purpose we will make the work known throught the industry, work with the IETF and the W3C to develop standards, and we will make code available to developers as free and open source software.

2 Making the Work Known

We will explain the benefits of our approach and discuss the technical details by:

1. Initiating and participating in discussions on mailing lists such as the Id-Commons mailing list, the identity mailing list of the W3C, and the mailing lists of the Web Security and TLS working groups of the IETF.

2. Writing blog posts and white papers and announcing them on Twitter.

3. Participating in industry meetings, such as the Internet Identity Worshop, the NIST IDtrust workshop, the quarterly IETF meetings, and relevant W3C workshops.

4. Joining industry organizations such as Kantara [74] and OIX [75].

3 Developing Standards

Standards will be needed for the certificate renewal protocol, for credential requests to the browser by the relying party, and for the new TLS phase that will be used by the browser to present multiple credentials of multiple types to the relying party. We assume that an extension of keygen for automated certificate issuance will have already been developed by the W3C; we will work on that standard as part of the redirect-to-issuer project, or with our resources if that project is not funded.

3.1 Certificate Renewal

A standard will be needed to specify how the browser requests renewal of a certificate, and how the certificate issuer responds to the request by sending a new issuance date, a new expiration date and a new signature which the browser uses to construct the renewed certificate, with confidentiality and integrity protection of the request and the response using shared symmetric keys. Also a further extension to the functionality keygen will be neeed within the HTML5 standard, so that shared keys can be established when the certificate is first issued.

We plan to initiate the discussion on certificate renewal at the W3C, taking advantage of the existing identity mailing list and an anticipated workshop on the use of public key certificates on the Web. The W3C works closely with the IETF. After the initial discussions at the W3C, it will be decided where the standardization work should take place. While the further keygen extension should be specified by the W3C as part of the HTML5 standard, the renewal protocol could well be developed by the IETF.

3.2 Credentials Request

Standards will also be needed for allowing the relying party to request credentials from the browser. We envision a simple syntactic standard specifying how a request is encoded using HTTP headers, and sparate standards specifying the semantics of the various OIDs that designate credential classes.

The syntactic standard will specify an HTTP header defining a single credentials requirement. An HTTP response message may contain multiple instances of the header to specify multiple requirements, all of which must be met. A requirement may consist of one requested credential class, or a list of alternative credentials classes, requiring the browser to present a credential from one of the classes. Each credential class will be specified by an OID, with optional parameters to specify which attributes must be disclosed for credentials that allow for selective disclosure.

We believe it is best to leave the specification of the OID-semantics standards to the communities that use the credential classes designated by the standards. The semantics of an OID will specify the format and contents of a credential in the corresponding class, and how trust on the credential is established. As an example, we will engage with at least one particular community and develop a standard for the semantics of a credential class to be used by that community. Candidate communities and credential classes include:

- Providers of personal data storage services [73], which will provide personal data certificates.

- Email service providers, which will provide email address certificates.
- Credit card issuing banks and shopping cart software providers, which will issue and accept credit card certificates respectively. (Notice that a credit card certificate provides strong security for a credit card transaction but does not change the payment processing; hence acquiring banks need not be involved, except in reducing the credit card fees charged to merchants who take advantage of the stronger security.)

3.3 TLS Extension

A standard will be needed to specify a TLS extension that will introduce a new phase for presentation of multiple credentials by the TLS client to the TLS server, the new phase following the TLS handshake so that it enjoys confidentiality and integrity protection. Additional standards will be needed for each type of credential that may be presented during the new phase.

We will propose to the IETF the development of a standard for the new TLS phase and a separate standard for submitting a public key certificate backed by a certificate chain and proving knowledge of the corresponding private key during the new phase. Standards for presentation of other kinds of credentials may be presented later.

Following normal IETF procedure, we will write two Internet drafts with initial versions of the standards and ask the TLS working group to take up further development. Since the proposed extension is a major one, it may require a rechartering of the working group. We will therefore also discuss our proposal with the Security Area Director.

4 Contributing Code as Free and Open Source Software

As indicated in the Data Management Plan, we will publish all the code that we write for the project so that our results can be verified by other researchers. Some of that code is proof-of-concept code only useful for demonstrating the results of the project; other code is reusable code, comprising software libraries and extensions of open source software. To facilitate industry adoption, as part of the Transitions phase, we plan to make the reusable code available to developers as free and open source software. That will require documenting and thoroughly testing the reusable code.

Our free and open source contributions will include:

1. An extension of Firefox supporting the keygen extensions, including the extension to issue certificates and import them into the browser (developed as part of the redirect-to-issuer project) and the further extension to set up shared keys for certificate renewal. We will contribute these extensions back to the Mercurial code repository of the Mozilla community [76].

2. Client and server libraries for the certificate renewal protocol. We will make those available on the Pomcor Web site.

3. An extension of Firefox that makes use of the client library for certificate renewal. We will contribute that extension to the Mercurial code repository of the Mozilla community.

4. An extension of NSS [70] implementing, on the client side, the new credential-presentation phase of TLS and the presentation of public-key certificates during that phase. We will contribute this back to the Mercurial code repository of the Mozilla community.

5. An extension of OpenSSL [72] implementing, on the server side, the new credential-presentation phase of TLS and the presentation of public-key ceritificates during that phase. We will contribute this back to OpenSSL.org.

6. An extension of mod_ssl [71] making use of the above OpenSSL extension to accept multiple credit card certificates during the new TLS phase and make the data in those certificates available to application code. We will contribute this back to Ralf Engelschall and/or publish it on the Pomcor Web site.

References

[1] National Strategy for Trusted Identities in Cyberspace. NSTIC Why We Need It. At http://www.nist.gov/nstic/NSTIC-Why-We-Need-It.pdf.

[2] Imperva. Military Password Analysis. Available at http://blog.imperva.com/2011/07/military-password-analysis.html.

[3] Imperva. Consumer Password Worst Practices. Available at http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf.

[4] Joseph Bonneau. Measuring Password Reuse Empirically, February 2011. Available at http://www.lightbluetouchpaper.org/2011/02/09/measuring-password-re-use-empirically/.

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

[6] 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.

[7] 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.

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

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

[10] A9.com. OpenSearch. At http://www.opensearch.org/Home.

[11] Pomcor. Noflail Search. A multisearch engine available at http://noflail.com/.

[12] F. Corella and K. Lewison. Searching the Web More Effectively With Multiple Simultaneous Queries. Presentation at the http://www.searchenginemeeting.net/2009/, available at http://pomcor.com/SearchingEffectively.ppt.

[13] F. Corella and K. Lewison. METHOD OF COMPUTING A COOPERATIVE ANSWER TO A ZERO-RESULT QUERY THROUGH A HIGH LATENCY API, April 2010. US Patent Application 20100100563.

[14] F. Corella and K. Lewison. FACILITATING BROWSING OF RESULT SETS, April 2010. US Patent Application 20100100836.

[15] 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.

[16] F. Corella and K. Lewison. PKAuth: A Social Login Protocol for Unregistered Apps, April 2011. Poster presented at the NIST IDtrust worshop, available at http://middleware.internet2.edu/idtrust/2011/papers/08-pkauth.pdf.

[17] F. Corella and K. Lewison. PKAuth: A Social Login Protocol for Unregistered Apps, March 2011. Whitepaper available at http://pomcor.com/whitepapers/PKAuth.pdf.

[18] Janrain. Social Login. At http://www.janrain.com/products/engage/social-login.

[19] NSTIC. Launch of the National Strategy for Trusted Identities in Cyberspace, April 15, 2011. At http://www.nist.gov/nstic/launch.html.

[20] Howard A. Schmidt. The National Strategy for Trusted Identities in Cyberspace and Your Privacy, April 26, 2011. White House blog post, available at http://www.whitehouse.gov/blog/2011/04/26/national-strategy-trusted-identities-cyberspace-and-your-privacy.

[21] Pomcor. Research in Web Technology. At http://pomcor.com/.

[22] National Strategy for Trusted Identities in Cyberspace. NSTIC NOI on Governance Comments. At http://www.nist.gov/nstic/governance-comments.html.

[23] F. Corella and K. Lewison. Pomcor's Response to the Notice of Inquiry On NSTIC Governance Structure, July 2011. Available at http://www.nist.gov/nstic/governance-comments/PomcorNOIResponse.pdf.

[24] Internet Policy Task Force Cybersecurity Green Paper. Comments received in Response to Federal Register Notice 110527305-1303-02. At http://www.nist.gov/itl/greenpapercomments.cfm.

[25] F. Corella and K. Lewison. Pomcor's Comments on Cybersecurity, Innovations and the Internet Economy, July 2011. Available at http://www.nist.gov/itl/upload/PomcorCybersecurityComments.pdf.

[26] NIST et al. 10th Symposium on Identity and Trust on the Internet, April 2011. At http://middleware.internet2.edu/idtrust/2011/.

[27] World Wide Web Consortium. W3C Workshop on Identity in the Browser, May 2011. At http://www.w3.org/2011/identity-ws/.

[28] F. Corella and K. Lewison. Nstic, privacy and social login, May 2011. Position paper presented at the W3C Identity in the Browser workshop, available At http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_48.pdf.

[29] IIW Wiki. IIW Notes, May 2011. At http://iiw.idcommons.net/IIW_12_Notes.

[30] Karen Lewison. How to meet privacy goals of NSTIC, May 2011. IIW 12 session notes, available at http://iiw.idcommons.net/How_to_meet_privacy_goals_of_NSTIC.

[31] IIW Wiki. IIW Notes, October 2011. At http://iiw.idcommons.net/IIW_13_Notes.

[32] Karen Lewison. Deployment and Usability of Crypto Credentials, October 2011. IIW 13 session notes, available at http://iiw.idcommons.net/Deployment_and_Usability_of_Crypto_Credentials_%28TH5K%29.

[33] Iana Bohmer. How do Different Technologies Align with the 4 NSTIC Guiding Principles, October 2011. IIW 13 session notes, available at http://iiw.idcommons.net/How_do_Different_Technologies_Align_with_the_4_NSTIC_Guiding_Principles_%28W4C%29.

[34] NIST. Meeting on Privacy-Enhancing Cryptography, December 2011. At http://www.nist.gov/itl/csd/ct/pec-workshop.cfm.

[35] Francisco Corella. Prospects for Using Privacy-Enhancing Technologies in the NSTIC Ecosystem, December 2011. Introductory slides for the panel on *Privacy in the Identification Domain* at the NIST *meeting on privacy-enhancing cryptography*, available at http://csrc.nist.gov/groups/ST/PEC2011/presentations2011/corella.pdf.

[36] Francisco Corella. Prospects for Using Privacy-Enhancing Technologies in the NSTIC Ecosystem, October 2011. Series of four blog posts, starting at http://pomcor.com/2011/10/04/pros-and-cons-of-u-prove-for-nstic/.

[37] The White House. The Comprehensive National Cybersecurity Initiative. At http://www.whitehouse.gov/cybersecurity/comprehensive-national-cybersecurity-initiative.

[38] World-Wide Web Consortium. Public Identity Mailing List. At http://lists.w3.org/Archives/Public/public-identity/.

[39] Harry Halpin. Email message expressing W3C interest in future work on certicates, October, 18 2011. Available at http://lists.w3.org/Archives/Public/public-identity/2011Oct/0003.html.

[40] Harry Halpin. Email message expressing personal interest in future work on certicates, November, 11 2011. Available at http://lists.w3.org/Archives/Public/public-identity/2011Nov/0032.html.

[41] Ian Hickson, Editor. HTML5, 10.5.16 The keygen element. At http://dev.w3.org/html5/spec/the-button-element.html#the-keygen-element.

[42] 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.

[43] 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. Available at
http://datatracker.ietf.org/doc/rfc5280/.

[44] J. Hoyt et al. OpenID Simple Registration Extension 1.0, June 2006. Available at
http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[45] D. Cooper, R. Housley, and S. Turner. An Internet Attribute Certificate Profile for Authorization, June 2010. Available at
http://tools.ietf.org/html/rfc5755/.

[46] N. Ragouzis et al. Security Assertion Markup Language (SAML) V2.0 Technical Overview, March 2008. Available at
http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf.

[47] Francisco Corella. Structured certificates and their applications to distributed systems.

[48] Francisco Corella. Method and apparatus for providing field confidentiality in digital certificates, October 2004. US Patent 6,802,002.

[49] Microsoft Corporation. U-Prove Home Page. At
http://www.microsoft.com/u-prove.

[50] 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.

[51] 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.

[52] J. Camenisch, P. Bichsel, and T. Gross. Idemix Blog. At
http://idemix.wordpress.com/.

[53] 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.

[54] Tolga Acar and Lan Nguyen. Revocation for delegatable anonymous credentials. In *Proceedings of the 14th international conference on Practice and theory in public key cryptography conference on Public key cryptography*, PKC'11, pages 423–440, Berlin, Heidelberg, 2011. Springer-Verlag.

[55] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. Solving revocation with efficient update of anonymous credentials. In *Proceedings of the 7th international conference on Security and cryptography for networks*, SCN'10, pages 454–471, Berlin, Heidelberg, 2010. Springer-Verlag.

[56] NIST. Personal Identity Verification (PIV) of Federal Employees and Contractors, March 2006. Available at http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf.

[57] Department of Defense. Common Access Card (CAC). At http://www.cac.mil/.

[58] ITU-T. X.660: Information technology Open Systems Interconnection Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the ASN.1 Object Identifier tree, August 2004. Available at http://www.itu.int/ITU-T/studygroups/com17/oid/X.660-E.pdf.

[59] T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2, August 2008. Available at http://tools.ietf.org/html/rfc5246.

[60] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 103–112, New York, NY, USA, 1988. ACM.

[61] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 186–194, London, UK, 1987. Springer-Verlag.

[62] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[63] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, June 1999. Available at http://tools.ietf.org/html/rfc2560.

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

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

[66] Matasano Security. Javascript Cryptography Considered Harmful. Available at http://www.matasano.com/articles/javascript-cryptography/.

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

[68] Anil John. SAML v2 Profiles for PIV Subjects and Backend Attribute Exchange, June 2009. Available at http://blog.aniltj.org/2009/06/saml-v2-profiles-for-piv-subjects-and.html.

[69] Don Thibeau. Open Attribute Exchange Network (OpenAXN). At https://sites.google.com/site/streetidentitylmnop/workinggroup.

[70] Mozilla. Network Security Services (NSS). At https://developer.mozilla.org/En/Developer_Guide/Source_Code.

[71] Ralf S. Engelschall. Home page of the mod_ssl project. At http://www.modssl.org/.

[72] OpenSSL. Home page of the OpenSSL Project. At http://www.openssl.org/.

[73] Personal Data Ecosystem Consortium (PDEC). At http://personaldataecosystem.org/.

[74] Kantara Initiative. At http://kantarainitiative.org/.

[75] Open Identity Exchange (OIX). At http://openidentityexchange.org/.

[76] Mozilla. Working with Mozilla source code. At https://developer.mozilla.org/En/Developer_Guide/Source_Code.