Security Assurance Levels for Crypto in the Browser

BJ Perng, Egistec, bj.perng@egistec.com
Arshad Noor, StrongAuth, arshad.noor@strongauth.com
18-July-2014 for the W3C Web Cryptography Next Steps Workshop

Previous and ongoing work in cryptographic protocols indicates a strong interest on the part of service providers / relying parties to understand the reliability and risk characteristics of the cryptographic and identity systems used by their customers for authentication. For some examples, see position papers from the previous W3C Workshop on Identity in the Browser (http://www.w3.org/2011/identity-ws/papers.html) including:

1. National Strategy for Trusted Identities in Cyberspace - Requirements and Potential Use Cases (by Peter Alterman of NSTIC/NIH)
2. Statement of Interest and Requirements for W3C Workshop on Identity in the Browser (by Dan Schutzer of Financial Services Roundtable/BITS)

A motivating factor behind the initial WebCrypto work was to provide the web platform access to more reliable cryptographic primitives than could be bootstrapped simply from JavaScript libraries, by delegating these functions to implementations in the core browser, OS or even hardware. These improvements in cryptographic reliability are good for users, but unless they can somehow attest to this reduction in risk to the services and applications they interact with, it will not allow those providers to build new services that take full advantage of the improvements.

To the extent that proposals for Next Steps in Web Cryptography envision explicitly connecting hardware tokens to the Web, the need for remote services to be able to understand the security profile they provide (as compared to a software only implementation) is even more critical to unlocking meaningful new use cases.

These characteristics are distinct from, e.g. the strength of the proofing process associated with an identity that is being asserted as part of an authentication, the resistance to attack of a particular protocol construction, or the level of assurance provided by a multi-factor authentication system in which a hardware crypto element is only a single component.

In order for web crypto APIs to be “universal” for both authentication (as contrasted to the current situation where a user often has a different multi-factor device for every relying party) and as general-purpose cryptographic components, a standardized way must be provided describe relevant security information about their implementation characteristics,

including, but not necessarily limited to:

1. Key protection methods (e.g. HSM, software, trusted execution environment)
2. User verification methods integrated with key protection (e.g. fingerprint readers or PIN pads)
3. Strength and reliability characteristics for each, including possibly third-party certifications

Furthermore, these must be conveyed with some meaningful cryptographic attestation in order for them to be relied upon.

Example Approach

The FIDO Alliance UAF protocol (http://fidoalliance.org/specifications/download) has taken one approach towards a solution to this problem.

In UAF, an "authenticator" is a logical unit of functionality that packages cryptographic operations, key storage and user verification functions. The following features are then provided:

1. Each model of authenticator is assigned an identifier string
2. Authenticators can report limited metadata about themselves directly to web applications through a JavaScript API
3. Authenticators can provide a cryptographic attestation of their identity as part of conveying a public key to a remote relying party
4. If it desires to verify this attestation, the remote relying party can go to a trusted third party and look up valid attestation keys and additional, more detailed metadata about the authenticator.

Two example cases are provided here as a starting point of discussion.
Two existing schemes, NIST/LoA[1] and IPF[2], have been exercised for the mapping.

SecAssurance	UAF Authnr Key Protection flags				UAF Authnr Matcher Protecton flags				Secure	NIST/	IPF
Case	SE	TEE	HW	SW	SE	On Chip	TEE	SW	Display	LoA*	Scale**
1		X				X			X	3	5
2	X				X				X	4	7
3…											

Case 1 is Fingerprint sensor with Authentication Key protected in TEE, Matching executed on chips, secure display available.

---

1 http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf

2 http://middleware.internet2.edu/idtrust/2008/papers/01-noor-ipf.pdf

Case 2 is Fingerprint sensor with Authentication Key protected in SE, Matching executed in SE, secure display available.

In the UAF protocol, the basic characteristics of the device would be available both through JavaScript APIs and in a detailed metadata statement. The third-party assertions such as the NIST LoA or IPF Scale, along with the device's attestation public keys, would only be listed in the detailed metadata statement.

We believe that capabilities of this type, and standardized, extensible descriptors of relevant characteristics, are essential components for hardware components to reach their full potential in the WebCrypto landscape.