Proposal for a discovery and connection API for Proximity Security Devices

Philip Hoyer – Director Strategic Innovation - HID Global July 18th 2014

phoyer@hidglobal.com

Abstract:

In this paper we present a potential vision for an in browser API that would allow the discovery, capability listing and connection to proximity (NFC, BLE, etc) security devices (e.g. smart cards, BLE fobs,) that can then be connected to for the purpose of crypto operations or generation of authentication credentials in multi factor authentication, for example generation of a One-Time-Password (OTP) .

Premise and Rationale

In the latest generation of devices that host the browser a new set of proximity transport interface technologies such as NFC and BLE (Bluetooth low energy) is increasingly implemented. These interfaces make it easy to connect from the device hosting the browser to other devices either other computing or mobile devices but also single purpose hardware such as location beacons, or credentials such as contactless smart cards, BLE fobs, Bluetooth connected fingerprint readers etc which have security capabilities such as crypto functionality or OTP generation.

The contactless interface makes the user experience of the security devices very appealing (simply tap a contactless smart card, or simply have a BLE fob ‘with you’ in range of the browser hosting device) which makes it really appealing from a security process perspective.

We also know from other standardisation efforts such as FIPS 201-2, ANSI/INCITS 504-1 GICS, that capabilities to utilize PKI certificates and credentials on the PIV card over contactless and the new Virtual Contact (Secure Channel protected contactless) interface is highly desirable.

Not only are there more and more dedicated hardware security devices that are based on the NFC/BLE technologies such as the new generation of HID physical Access Cards (HID iCLASS Seos) but due to the fact of Host Card Emulation in latest mobile phone operating systems, mobile based credentials also can use

these technologies for security purposes (e.g. tap your phone to the laptop, or have the phone in BLE range of the laptop)

It seems hence that is imperative to open the capabilities of these new contactless credentials to web applications via a browser based web API.

Proposal

Due to the contactless nature and the fact that some proximity technologies such as HFC only operate at extremely close ranges mean that the device on which the browser s running is in constant change of what proximity devices are in range. Hence the proposal to have a discovery API that would allow the web application to:

1. Know what proximity device are in range or maybe through previous interactions could be asked to be brought in range (e.g. please tap the contactless smart card you used last time)

2. Create a standardised API to retrieve the capabilities list of the device so that it can be mapped to the algorithms and crypto mechanism already defined in the W3C Crypto API (e.g. this smart card can do PKI sign with RSAES-PKCS1-v1_5 or AES or a capability to generate RFC4226 HMAC based OTP)

Let’s take the two most prevalent proximity interface technologies into consideration, NFC and BLE.

There are really different levels of abstraction here and one could build upon the other:

1. Low level communication oriented API, to be able to communicate with the device from a web based application – high priority / fundamental

    a. W3 has done some forays into NFC with the Web NFC API (http://www.w3.org/TR/nfc/) but it currently is limited to reading NFC forum tags and does not expose the natively present reading capability of the NFC controller to be able to communicate arbitrary Application Protocol Data Units (APDUs) with a contactless smart card for example. As would be needed to support utilization of a PIV 201-2 smart card or an HID Seos Access Card.

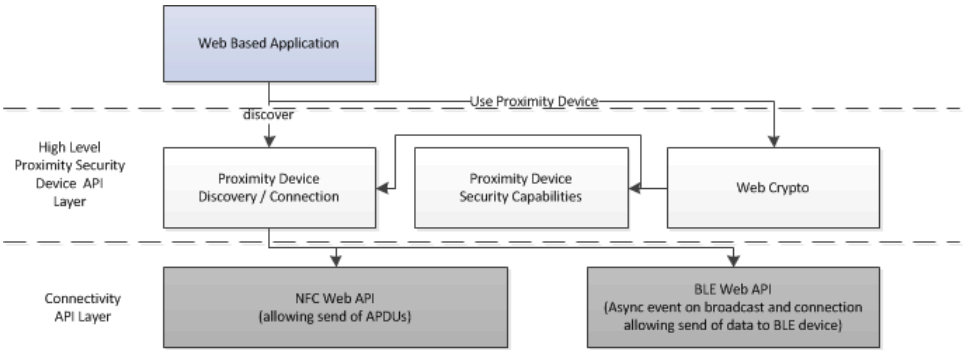
    b. There seems no current web API for BLE, so the proposal would be to have a Web API that would allow BLE communication (asynchronous wake up events from BLE broadcast and a connection oriented API to be able to connect to the BLE and exchange data with it)

2. A Higher level API that will use the Low Level communication API: with the following functional areas:

    a. Discovery / Connection and listing of known Proximity Security devices independent of transport (abstract NFC / BLE, etc as much as possible)

    b. Retrieval of the security capabilities of known Proximity Security Devices

    c. Connection API to the security devices at an abstraction level that would then map it to the existing W3C Crypto API Level (e.g. ability to retrieve a handle to a SubtleCrypto interface from a connected device handle)



Conclusion

In the quest to get rid of passwords, contactless proximity interface based security devices present one of the best and convenient user experience. By enabling browser based application to leverage them for security (authentication and crypto function) processes through a set of layered Proximity Device APIs we will see new generation of user convenient and secure web applications. We feel that this proposal is ideally aligned and builds on the initial work of the W3C Crypto API charter.