XTEC COMMENTS TO IAB RECOMMENDED REVISIONS TO FIPS 201 Version 1.1

Introduction

Overall, the document appears to be an improvement over the original document. With a few changes, this seems to be a good position to get all agencies to give a consolidated position on the FIPS 201 document. One thing to note is the changes to this document place the real risk on the outcome of the SP 800-73 document. Besides the changes highlighted in the document, I would breakdown the changes into three levels: critical to change, nice to have, or no real consequence. Our comments are based on technical merit, with limited comments on policy/business process issues.

Critical areas to have changed:

- Section 8.2.1 Rapid PIV Authentication should be allowed with either asymmetric or symmetric technology. Besides the speed of the transaction of using asymmetric or symmetric technology, contactless cards have been in use for a long time using symmetric keys for authentication. There are hundreds of millions of cards currently deployed with this capability. On the other hand, contactless cards are just starting to be tested with asymmetric technology. We do not know for sure if this will have any negative affects. Example would be differential power analyst. It took years for contact cards to over come this attack. More studies need to be done to ensure the additional power needed to perform asymmetric keys would not be vulnerable to this type of attack. Requiring cards to contain at least one symmetric key on the contact and contactless chip, if only as a backup or option use for the agency, would give cards a proven fall back insurance policy. What happens if the asymmetric key is every compromised? Every card natively supports the symmetric keys, so there would be no cost impact to the issuing agency. This would also future interoperability by allowing agencies such as GSA and State Department, who have already deployed the PACs High Security Profile with symmetric keys, to not have to reduce their security requirements to be interoperable with other agencies.
- SEIWG is a magnetic stripe standard. It has been used in smart cards as a transition until an access control standard using the chip could be defined. The limitations of SEIWG are well defined. There have been many attempts to keep backwards interoperability, which is a good thing. One area where this should be explored is how to map the legacy SEIWG data into the CHUID 16 byte GUID. This would allow a smooth transition to the use of the GUID, and can eliminate the problems of SEIWG.

Nice to have changed:

- If each card were required to contain in the CHUID Container the FASC-N, GUID, Authentication Key Map, and Asymmetric Signature, interoperability would be easier to achieve. This would allow each agency to decide which level of security to implement at each access control point. It would be the requirement of the agency to determine how to get the information into their system, therefore making key management a function of MOUs in the short term while a larger scale system can be developed.
- It is the belief that Certicom holds the patent for Elliptic Curve. If this is to be made a requirement, I would highly recommend a legal decision if made prior and Certicom is required to sign a waiver over any legal challenge to this claim. I am not aware of any

federal programs currently using Elliptic Curve, this could put an unnecessary requirement on deployments.

- From a cryptographic standpoint, three key triple DES does not makes sense. Very few products, if any, support this. NIST itself recommends not using it sense the benefit over two key is only one bit of entropy. Instead, we should move from two key 3DES to AES.
- SHA has come under some scrutiny recently. It was not too long ago MD5 was recommended as the Hash Algorithm. We should have some flexibility here incase there is any flaws found in SHA in the future. The standard needs to be more future proof in this area.
- The requirement for readers to conform to PC/SC is not required. Many readers will not work on a PC, such as physical access readers. This standard was optional in GSC-IS and should still be.

No real consequences from a technical standpoint:

- Agency CSN and Issuer Id Number should be optional. They do not add to the security of the card.
- Section 8 Graduated Criteria – this could be simplified by having the following levels of security
  - Level 0 – Card is used as flash pass. Card can contain visual security features
  - Level 1 – Card is used with no data or card authentication (similar to PACS Low assurance)
  - Level 2 – Card is used with verified/signed data (similar to PACS Medium)
  - Level 3 – Card is used with card and data authentication (similar to PACS High)
  - Level 4 – Card is used with level 3 security features and also requires either a PIN or Biometric to verify the card holder
  - Level 5 – Card is used with level 3 security features and also requires both a PIN and Biometric to verify the card holder
- All other red line contained in the markup.

Backup information

One of the fundamental axioms of credential authentication is that it requires a secret key embedded on the card. It doesn't so much matter what infrastructure you use to put the key there, what is important is that the key on the card is secret and must never be revealed. This is true regardless if you use SKI or PKI.

Since the key can never be revealed, it follows that the only way to authenticate the smart card is through a challenge-response mechanism: give the card a random number challenge and the card returns a cryptogram. To validate the cryptogram you must calculate the cryptogram yourself using a secret key that matches the secret key on the card.

If you are using PKI, this could be a public key, but the issue with PKI is that it is too slow for PACS and contactless cards do not support it. Remember we need to build something that can work with both contact or contactless cards. Symmetric key cryptography is the better solution in these circumstances. But, again, the key must not be revealed, so the key is always maintained in hardware. In practice the hardware is much like the smart card, that is, a tamper proof microchip that can duplicate the challenge response of the smart card. Give the chip the same challenge and you get back the same cryptogram. If the cryptograms match, the keys are the same, and card has been authenticated.

So now to the point: how do you get the keys in the card and into the microchip securely? And, the second question, how do you make every key different to reduce the risk to the global system, and still perform authentication?

The question of key insertion is a matter of what happens at card issuance. At card issuance, you have physical possession of the card. It is your hardware/software system that is injecting the key and so you trust it. Once the key is on the card, it is protected by the card operating system and cannot be discovered short of extraordinary effort, and so users of the card can trust that the key is authentic.

Putting the key in the authenticating microchip is only a little more difficult. The difficulty arises because the authenticating microchip is generally manufactured by a third party, whom you must trust with possession of your key.

One of the prime means to reduce your risk is to use key derivation. Key derivation means that the key you use for authentication has been cryptographically derived from the combination of a parent key and some small piece of data. The derived key is placed on the card with the piece of data. The cryptogram is calculated using the derived key, and is passed back along with the piece of data used to derive the key. The challenge and the piece of data are given to the authenticating microchip, which uses the combination to calculate the authenticating cryptogram.

In this scenario, the authenticating microchip still contains the parent key; however, key derivation comes to the rescue again. It is not necessary to have a single parent key. Key derivation can be thought of as a hierarchy of parent keys, each serving as a root key to its offspring.

For the GSA interagency card authentication methodology (in use by DOS, GSA and TWIC), each participating agency has its own key derived from the root key. In fact, there are many derived keys for each agency, and each derived key can be used to generate offspring keys that are used to authenticate the tokens. The reasoning being that an agency can reduce its risk by placing multiple derived keys on the card, but only a single derived key--which is not the root key--in the authentication device. The same key does not have to be given to every authentication device manufacturer.

This method works well for global systems that must maintain their own security yet interoperate with other entities outside their domain. But, for some more homogeneous organizations even this diversification is not enough. DoD, for example, is looking at a scenario where no root keys are stored. This means that each uniquely derived card key is kept in the authentication device.

In practical terms, not every key must be maintained. For PACS, only the keys of the persons expected to pass through the door must be kept. This could conceivably be a large number, but not beyond the capabilities of a PACS. These keys are kept in the access control panel, which is generally capable of holding 30,000 or more identifiers. The largest capacity I have heard of is 500,000, but I am not sure any door has every had some many people assigned to it.

To maintain security, each key is stored encrypted. The panel contains an authentication microchip that holds the panel-unique transport key. To perform authentication, the encrypted key and challenge are passed to the microchip, which decrypts the card key, computes the cryptogram, and passes back the result for authentication.

To enable this scenario, a registration process is used. Each card to be used in the PACS is registered to that panel by downloading the card key from a central database (use SSL or some other secure channel). The key is first derived from the card data stored in the database, then encrypted with the panel's transport key before being sent. This is similar to the way most PACS operate with the addition of token authentication.

One bonus of key derivation is that you get to authenticate the data used to derive the key at the same time you authenticate the token. This is a direct result of the challenge response. The data used to calculate the cryptogram must be the same as the data used to derive the key or else the cryptograms will not match. This is how the pin is handled. It can also be extended to the card expiration data. You can have multiple keys on the card for various combinations of data (i.e. card only, card and pin, card/pin/expiration data, etc).

As for checking for revocation, it's not the card that gets revoked, but the usage. To actually revoke the card you have to regain possession of the card. So what needs to be done is to notify all the places that the card can be used that the token is no longer valid, that is, cannot be used.

For PACS, usage revocation is relatively straight forward, since the user privileges can be rescinded from the database. Several conventions are available to distribute the RCL, from direct connection to email, depending on the degree of outside connectivity. The LEO system uses an innovative national paging system for instant global wireless revocation. Generally a

MOA is signed between the parties who allow usage of the card that designates the means of notification.