PKI (CAK) – Enabled PACS with PIV Card PACS Lessons Learned and Need for Speed

Prepared by: Bob Gilson and Tim Baldridge

Date: 03.03.2015

Briefing Goals

- DoD Use of Secure Messaging for Transit
- The “Key” Point
- Mandatory Security Check
- Transaction Optimization
- Additional Considerations
- Conclusion

DoD Future use of Secure Messaging for CAC high speed contactless use case requirements (OPACITY)

- What is OPACITY?
  - Secure Messaging (SM) is a simplified profile of OPACITY a secure open protocol as specified in ANSI/INCITS 504, initially developed from a DoD request for a contactless secure channel
  - SM is included in the next gen CAC profile for secure channel establishment over the contactless interface for all PIV uses
  - In a Transit fare pass use case SM may achieve high-speed application data transaction speeds under 300 ms
  - SM requires Elliptic Curve Cryptography (ECC) based Diffie-Hellman key exchange (keys are smaller than RSA)
  - SM may be implemented in an application on a CAC/PIV or as a profile supported by a ANSI/INCITS 504 platform
  - SM is an OPACITY Zero Key Management (ZKM) profile meaning no persistent binding. Future amendments to ANSI/INCITS 504 will mature capabilities for full secrecy

DoD Future use of Secure Messaging for CAC high speed contactless use case requirements (OPACITY)

- Where do we envision using it?

  Initially: Transit metro rail service access

  - METRO Access/Payment, transit Benefit for CAC/PIV card and potentially mobile NFC environments.
  - High throughput low risk.
  - **Potentially High volume perimeter DoD PACS Access.**
  - Much better than either free read or flash pass.
  - **Future potential contactless limited MCC retail payments, possibly DoD Travel card.**

- Co-exists on CAC with CAK for interoperability.

DoD Future use of Secure Messaging for CAC high speed contactless use case requirements (OPACITY)

- What has DoD learned so far with Opacity/PACS Test?

Used Opacity PACS configured reader and test CAC/PIV.

- Implemented Opacity contactless security, with all crypto checks turned off and persistent binding incorporated as baseline.
- Was lightning fast,~100ms.
- Need selective crypto checking at POST.
- Believe we need Opacity to run at the OS level, not on JAVA.
  - ✓Pilot testing required to confirm.
- Believe we also need next gen chip speeds, 2x-3x faster as well.
  - ✓Pilot testing required to confirm.

DoD Future use of Secure Messaging for CAC high speed contactless use case requirements (OPACITY)

- When?

Initial Proof of Concept, projected for late summer, 2015 with WMATA in NCR/volunteer riders/selected metro stations.

- Looking for other agency volunteers to join DoD and WMATA.
  - ✓Will take transit benefit coordination with current WMATA Transit Benefit interface processor.

The "KEY" Point

- Proof of Possession – The "HAVE" Factor is established through the use of approved cryptographic algorithms and protocols to verify the possession and control of a private "KEY" issued to a cardholder
- The association of a "KEY" to a cardholder is established by a trusted Issuer by binding the "KEY" to a cardholder identifier (UUID) in a certificate signed by the issuer
  - For CAK – x.509 Certificate
  - For SM – Card Verifiable Certificate (CVC)

Mandatory Security Checks

- Within the transaction, two critical cryptographic validations must always be performed
  1. The “Host -> Card -> Host” crypto must validate
  2. The signature on the certificate provided by the Card must validate before the host accepts the UUID from the certificate for access or the host must pre-register each individual card public key
- Definition: Endorsement Key
  - The public key used by the host to validate the digital signature of a certificate.
  - For SM see 3.3.7 Secure Messaging Certificate Signer
  - For CAK it is the PK of the superior X.509 Certificate

Transaction Optimization

- The fundamental basis of a solution architecture for sub-second performance is a closed validation strategy at the access transaction.
  - All information to complete the cryptographic validations is available at the edge within allocated time
  - Credential revocation checks are not synchronous with access transaction - they may be performed periodically
  - The “Endorsement Key” is local to the access point and is retrieved by direct or indirect indexing based on the UUID
  - As an alternative to the “Endorsement Key” each allowed associated card public key may be available at the edge – this approach may be limited to smaller populations

Transaction Optimization (cont)

- These optimizations infer a registration process that occurs before the access transaction which must include appropriate periodic cardholder revalidation
- Requirements for initial and periodic validation of an issuer asymmetric public Endorsement Key are based on the distribution method and Issuer practice, a single Endorsement key may be valid for millions of cards
- Requirements for initial and periodic validation individual cardholders are referenced to the UUID and based on associated information obtained at time of registration.

Additional Considerations

- There are two card performance metrics which are critical, 1) time for the FIPS 140 Power-Up Self-Test (POST), and 2) computational time for the GENERAL AUTHTENTICATE response ADPU
- ADPU exchange time and Host processing time must be efficient and complete within the remaining time. To accommodate a long Host latency, obtain and transmit the UUID to the host before beginning the card crypto
- ECC allows for smaller key objects that are important at lower transmission rates. SM requires ECC and for CAK ECC use is optional, however when used achieves a majority of the SM performance gain.

Conclusion

- The PIV PKI Card Authentication Key (CAK) is the only interoperable one-factor strong cryptographic solution
  - Now Mandatory, Beginning with FIPS 201-2
  - Provides full PKI-based revocation checking
  - With ECC CAK Key and ECC CAK issuing CA approaches cryptographically equivalence to SM timing when the same transaction optimizations are implemented

Questions?

Gilson, Irving R (Bob) irving.r.gilson.civ@mail.mil

Baldridge, Tim W tim.w.baldridge.civ@mail.mil

Backup Slides

PIV Card Authentication Evolution

- CAK Introduced in initial SP 800-73, April 2005
- UUID feature for NFI Cards in SP 800-73-3, Feb 2010
- Second Public Draft SP 800-73-4, May 2014
  - Asymmetric Card Authentication Key Mandatory
  - Card UUID Mandatory
  - Introduced Optional Secure Messaging (SM)

SP 800-73-4 2nd Public Draft
B.1.3 Authentication Using Card Authentication Key

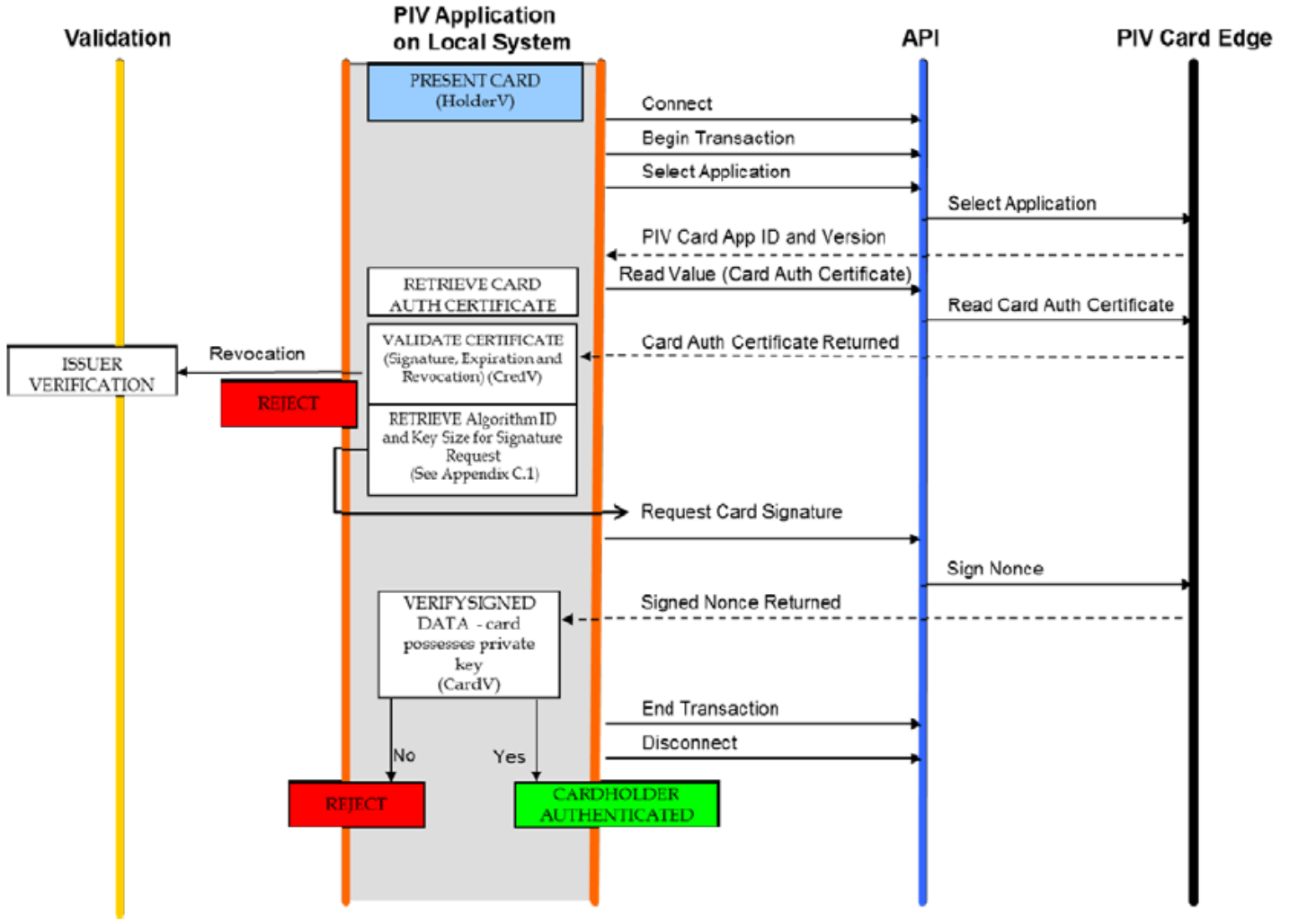
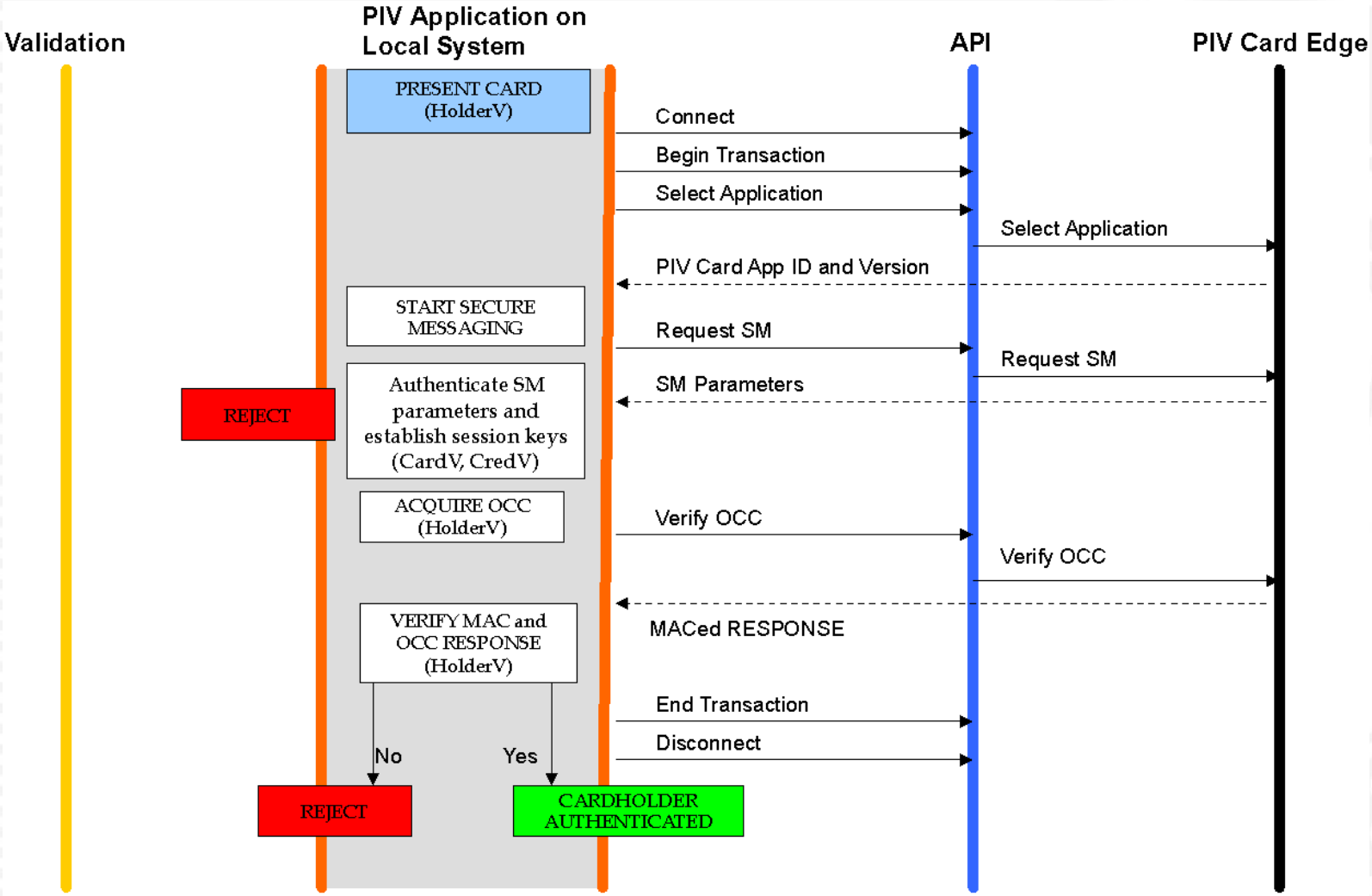


Figure B-4. Authentication using an asymmetric Card Authentication Key

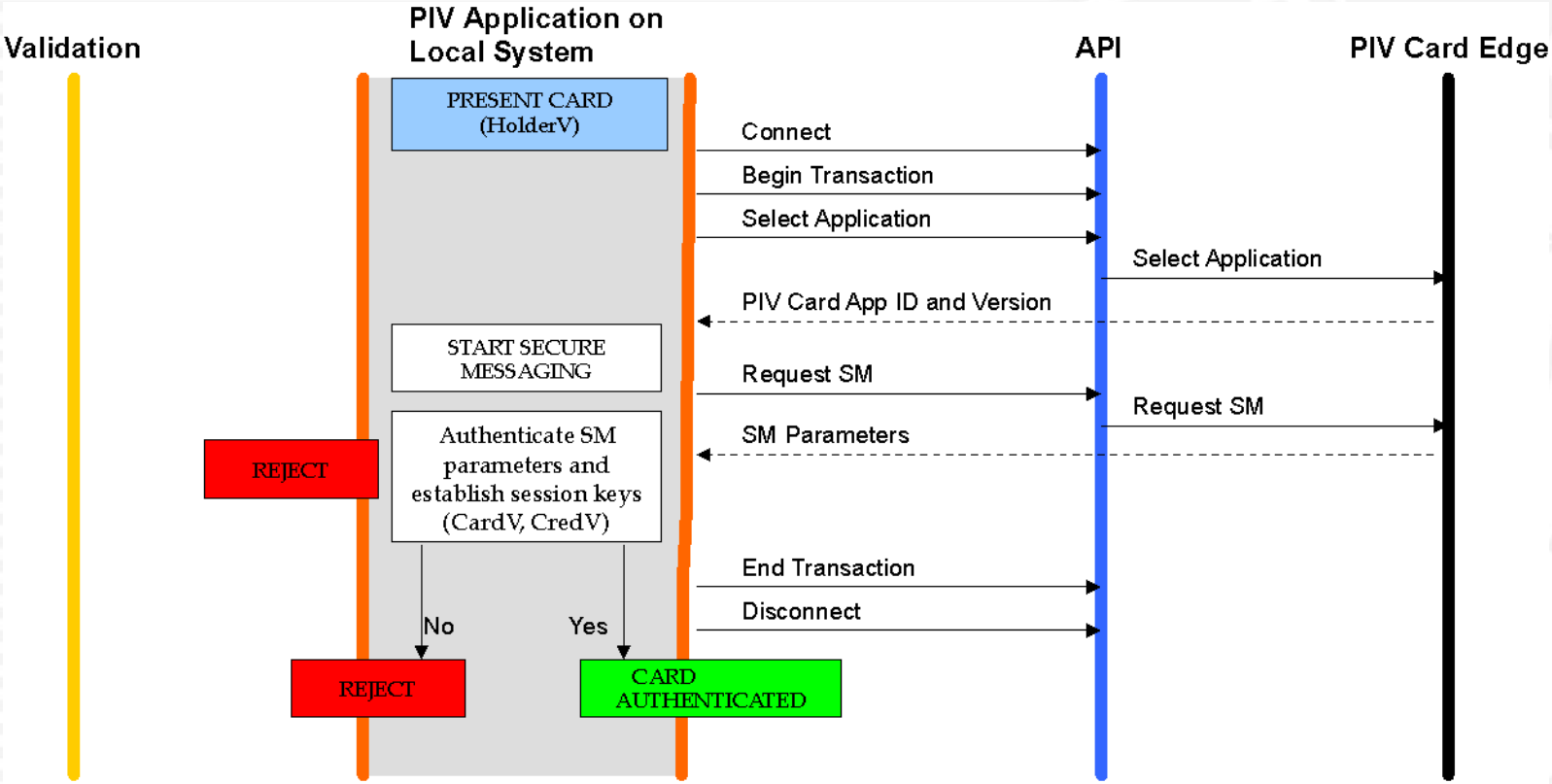
OPACITY and PIV

- For PIV, Secure Messaging (SM) a simplified profile of OPACITY with Zero Key Management - ZKM (ANSI 504-1) is described in SP 800-73-4 2nd Public Draft, Part 2, Section 4.1 Key Establishment Protocol
- SM is included for PIV for the purpose of protecting eavesdropping attacks on contactless PIV operations
- As a collateral benefit SM may be also be used for Card validation, like the Card Authentication Key (CAK)
- OPACITY is designed as an efficient and secure protocol based on Elliptic Curve Cryptography (ECC)

SP 800-73-4 2nd Public Draft
B.1.4 Authentication Using OCC (OCC-AUTH)



Authentication using SM (SM-Auth)



- Similar to OCC-Auth without Card Holder Validation
- One less transaction card-edge transaction than authentication with CAK

Transit Fare Token Requirement

- *Strongly resistant to tampering and counterfeiting*
- *Can be rapidly authenticated electronically*
  - *Fast Transaction time ~ 300mS (0.3 Seconds)*
- *Issued by providers whose reliability has been established*

- Fare collection system moves from stored value cards to a rider based back end account that is associated to a token presented at the fare gate

SM Transaction Basics

1. Host Resets Card and waits for Card to complete Power-On Self-Test and send Answer to Reset (ATR)
2. Host Selects PIV Application and Card Replies
3. Host generates an ephemeral key pair and initiates OPACITY ZKM protocol exchange with Card.
4. Card replies with AuthCryptogram and CVC.
5. Host validates both the AuthCryptogram and CVC, if valid then extracts the UUID from the CVC for access