



Cross-Agency Authentication using PIV Symmetric Keys

Dr. Sarbari Gupta

sarbari@electrosoft-inc.com

703-437-9451 ext 12

NIST Key Management Workshop

June 9-10, 2009



Overview

- **Personal Identity Verification (PIV) Cards contain optional symmetric keys**
- **We present a strong, rapid (and potentially two-factor) authentication scheme using PIV symmetric keys**



HSPD-12 and FIPS 201 Background

- **Homeland Security Presidential Directive 12, issued 08/2004:**
 - **Requires secure and reliable identification (for Federal employees & contractors) that:**
 - Is strongly resistant to identity fraud, tampering, counterfeiting, and terrorist exploitation
 - Can be rapidly authenticated electronically
- **FIPS 201 establishes the standard for Personal Identity Verification (PIV) Cards**

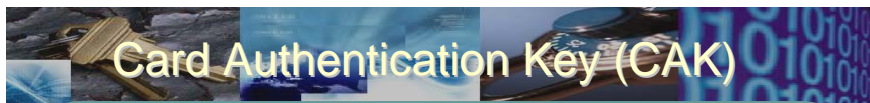


PIV Card Credentials

- **Mandatory Credentials:**
 - Cardholder Unique Identifier (CHUID)
 - PIV Authentication Private Key and X.509 Certificate (PKI)
 - Cardholder Fingerprints in Biometric Object (BIO)
- **Optional Credentials:**
 - **PIV Card Authentication Key (CAK)**
 - PIV Digital Signature Private Key & X.509 Certificate
 - PIV Key Management Private Key & X.509 Certificate
 - Cardholder Facial Image

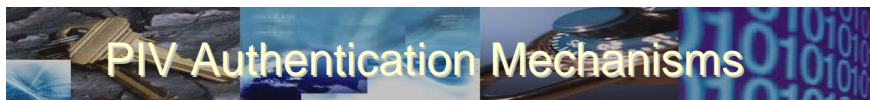


Pictures courtesy www.fedidcard.gov



Card Authentication Key (CAK)

- **According to FIPS 201-1, the CAK:**
 - Is Optional
 - May be symmetric or asymmetric
 - Is designed to support physical access applications
- **SP 800-116 strongly recommends that the CAK:**
 - Be included in every PIV Card
 - Comprise an Asymmetric Key and a certificate
- ***However, a symmetric CAK is permitted***



PIV Authentication Mechanisms

- **VIS** – Visually inspect the Card and match holder's face to picture on card
- **CHUID** – Validate signed CHUID (optional). The PIN is not required.
- **CAK** – Use the Card Authentication Key in a challenge response protocol. The PIN is not required.
- **BIO** – Match cardholder's fingerprint sample(s) to signed biometric data element without human attendant in view. The PIN is required to activate the card.
- **BIO-A** – Match cardholder's fingerprint sample(s) to signed biometric data element in view of human attendant. The PIN is required to activate the card.
- **PKI** – Use PIV authentication private key in a challenge response protocol that can be validated using the PIV authentication certificate. The PIN is required to activate the card.

PIV Authentication Mechanism Comparison

Authentication Mechanism	Contact?	Contactless?	Rapid?	Electronic?	# of Authentication Factors	Card is authentic (CardV)?	Credential is untampered (CredV)?	Card belongs to Holder (HolderV)?	Notes
VIS	-	-	N	N	2	Y (weak)	Y (weak)	Y (weak)	
CHUID w/o Sig Ver	Y	Y	Y	Y	1	N	Y(weak)	N	Expiration Date check
CHUID w/ Sig Ver	Y	Y	N	Y	1	N	Y	N	
BIO w/o Sig Ver	Y	N	N	Y	1	N	Y(weak)	Y(weak)	Expiration Date check; Gummy finger use possible
BIO w/ Sig Ver	Y	N	N	Y	1	N	Y	Y(weak)	Gummy finger use possible
BIO-A w/o Sig Ver	Y	N	N	Y	2	N	Y(weak)	Y	Expiration Date check
BIO-A w/ Sig Ver	Y	N	N	Y	2	N	Y	Y	
CAK - Symmetric	Y	Y	Y	Y	1	Y	-	N	
CAK - Asymmetric	Y	Y	N	Y	1	Y	Y	N	
PKI	Y	N	N	Y	2	Y	Y	Y	

Authentication using Symmetric CAK

- **GOAL: Cross-agency use of symmetric CAK for authentication to PACS**
 - Fulfill HSPD-12 vision of reliable, strong, rapid authentication across Federal government
- **Difficulties with Symmetric CAK authentication:**
 - Typical symmetric key Challenge-Response schemes require CAK to be known by Verifier
 - Cross-Agency Verifier will not know CAK
- **Advantages of Symmetric CAK use:**
 - Strong authentication (compared to CHUID)
 - Very rapid
 - Does not require activation with PIN
 - Can be performed over contactless interface

CAK Authentication V1 - Enrollment Step

- **Establish Trust in PIV Card**
 - **Visitor Cards:** Verify PIV Card and Holder using PIV Auth Key
 - **Local Cards:** Issue PIV Card in a compliant manner
- **Retrieve FASC-N from PIV Auth Certificate**
- **Send Challenge C to Card and request CAK operation**
 - Challenge C = HASH [FASC-N, Nonce]
 - Collect Response R = ENCRYPT {C, CAK}
- **Store the following in a PACS DB**
 - FASC-N
 - Nonce
 - H-Resp, where H-Resp = HASH [R]

CAK Authentication V1 - Authentication Step

- **Read CHUID from PIV Card**
 - Retrieve FASC-N from CHUID
- **Lookup PACS DB using FASC-N to obtain:**
 - Nonce
 - H-Resp
- **Send Challenge C' to Card and request CAK operation**
 - Challenge C' = HASH [FASC-N, Nonce]
 - Collect Response R' = ENCRYPT {C', CAK}
- **If Hash[R'] == H-Resp, authentication successful**
- **Grant User access based on authenticated FASC-N**



Analysis

- **Context:**
 - PKI Authentication at Enrollment establishes basis for trust
 - Authentication in physical access environment
 - Compare to CHUID authentication
 - Very weak; Requires signature validation
- **Benefits:**
 - Authentication scheme is much stronger than CHUID
 - Very rapid (no asymmetric key operation)
 - Verifies FASC-N in CHUID (CredV)
 - Can be performed over contactless interface
- **Areas of Improvement:**
 - Single Authentication Factor
 - Challenge-Response pair is static
 - Can be easily captured and replayed later



CAK Authentication V2 - Enrollment Step

- **Establish Trust in PIV Card**
 - **Visitor Card:** Verify PIV Card and Cardholder using PIV Auth Key
 - **Local Card:** Issue PIV Card in a compliant manner
- **Retrieve FASC-N from PIV Auth Certificate**
- **Prompt User for P-PIN (NEW!)**
- **Send Challenge C to Card and request CAK operation**
 - Send Challenge: $C = \text{HASH} [\text{FASC-N, Nonce, P-PIN}]$ (NEW!)
 - Collect Response $R = \text{ENCRYPT} \{C, \text{CAK}\}$
- **Store the following in a PACS DB**
 - FASC-N
 - Nonce
 - P-PIN Use Flag (TRUE/FALSE) (NEW!)
 - H-Resp, where $\text{H-Resp} = \text{HASH} [R]$

CAK Authentication V2 - Authentication Step

- **Collect User CHUID from PIV Card**
 - Obtain FASC-N from CHUID
- **Lookup PACS DB using FASC-N to obtain:**
 - Nonce
 - P-PIN Use Flag (TRUE/FALSE) (NEW!)
 - H-Resp
- **If P-PIN Flag is TRUE, prompt User for P-PIN' (NEW!)**
- **Send Challenge C' to Card and request CAK operation**
 - Send Challenge: $C' = \text{HASH} [\text{FASC-N, Nonce, P-PIN}']$ (NEW!)
 - Collect Response $R' = \text{ENCRYPT} \{C', \text{CAK}\}$
- **If $\text{Hash}[R'] == \text{H-Resp}$, authentication successful**
- **Grant User access based on authenticated FASC-N**

CAK Authentication V3 - Enrollment Step

- **Establish Trust in PIV Card**
 - Visitor Card: Verify PIV Card and Cardholder using PIV Auth Key
 - Local Card: Issue PIV Card in a compliant manner
- **Retrieve FASC-N from PIV Auth Certificate**
- **Prompt User for P-PIN**
- **Send Challenge C to Card and request CAK operation**
 - Send Challenge: $C = \text{HASH} [\text{FASC-N, Nonce, P-PIN}]$
 - Collect Response $R = \text{ENCRYPT} \{C, \text{CAK}\}$
- **Store the following in a PACS DB**
 - FASC-N
 - Nonce
 - P-PIN Use Flag (TRUE/FALSE)
 - H-Resp, where $\text{H-Resp} = \text{HASH} [R]$

CAK Authentication V3 - Authentication Step

- **Collect User CHUID from PIV Card**
 - Obtain FASC-N from CHUID
- **Lookup PACS DB using FASC-N to obtain:**
 - Nonce
 - P-PIN Use Flag (TRUE/FALSE)
 - H-Resp
- **If P-PIN Flag is TRUE, prompt User for P-PIN'**
- **Send Challenge C' to Card and request CAK operation**
 - Send Challenge: C' = HASH [FASC-N, Nonce, P-PIN']
 - Collect Response R' = ENCRYPT {C', CAK}
- **If Hash[R'] == H-Resp, authentication successful**
- **Send Challenge C'' to Card and request CAK operation**
 - Send Challenge: C'' = HASH [FASC-N, Nonce'', P-PIN'] (NEW!)
 - Collect Response R'' = ENCRYPT {C'', CAK} (NEW!)
- **Update the following in the PACS DB using FASC-N (NEW!)**
 - Nonce = Nonce''
 - H-Resp = HASH [R'']
- **Grant User access based on authenticated FASC-N**

Summary

- **Symmetric CAK provides an alternative for rapid, strong authentication of PIV Card**
- **Can be coupled with a P-PIN to add a second factor of authentication**
- **Cached Response can be updated with each use to minimize exposure to replay attacks**
- **QUESTIONS??**
 - *“For those viewing via webcast, please submit questions for this presentation to kmwquestions@nist.gov”*