# Secure Messaging and On Card Comparison Companion Paper

FRTC VERSION 1.4.2 Rev A

## FIPS 201 EVALUATION PROGRAM

**March 31, 2021**

Version 1.0

Office of Government-wide Policy
Office of Technology Strategy
Identity Management Division
Washington, DC 20405

# Document History

| Status | Version | Date | Comment | Audience |
|---|---|---|---|---|
| Draft | 0.0.1 | 2/12/2021 | Initial draft for comment | Public |
| Initial Release | 1.0 | 3/31/2021 | Initial publication | Public |

# Table of Contents

# 1. Background

The GSA FIPS 201 Evaluation Program Lab for PACS tests ePACS systems according to NIST [SP800-73]. The test cases that support the APL are being updated to now include specifications from NIST SP 800-73-4. This version of the special publication added optional features for Secure Messaging (SM), and On-Card Comparison (OCC) protocols.

SM provides confidentiality and integrity of card transactions both contact and contactless, and specifically supports a Virtual Contact Interface in contactless mode. Additionally, SM is a prerequisite for the use of OCC due to the transmission of sensitive personal information (e.g., fingerprint biometric).

OCC provides a method of user private key activation using a provided biometric in place of a more traditional PIN input.

# 2. Objectives

This document aims to provide informal guidance to ePACS vendors considering implementation of SM and OCC capabilities within their products as detailed in NIST Guidance.

In support of this goal, the following chapters present a methodology and details on how to determine whether a presented credential supports SM or OCC-AUTH, and the associated test cases used to determine if a PACS solution is securely leveraging SM and OCC capabilities. Both of these factors should assist PACS vendors in engineering their solutions to leverage SM and OCC capabilities.

# 3. Methodology

## 3.1 SM/OCC-AUTH Decision tree

The decision tree necessary to confirm a PIV card supports SM/OCC-AUTH is shown in two formats. Figure 1 is a trigger based workflow. Section 3.2 provides a step by step checklist.

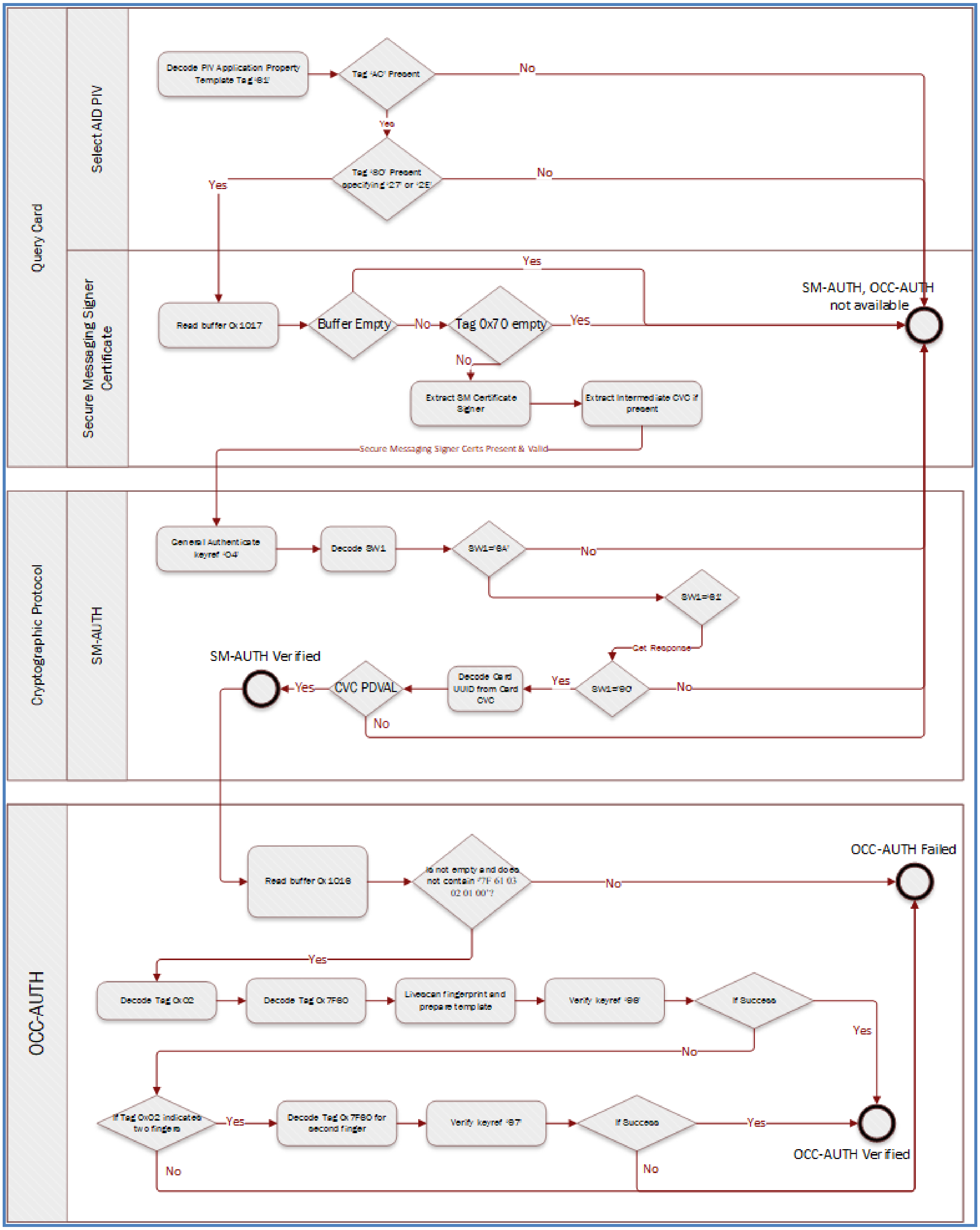

Query Card
Select AID PIV
Decode PIV Application Property Template Tag '61'
Tag 'AC' Present
No
Yes
Tag '80' Present specifying '27' or '2E'
Yes
No
Secure Messaging Signer Certificate
Read buffer 0x1017
Buffer Empty
No
Tag 0x70 empty
Yes
Yes
No
SM-AUTH, OCC-AUTH not available
Extract SM Certificate Signer
Extract Intermediate CVC if present
Secure Messaging Signer Certs Present & Valid
Cryptographic Protocol
SM-AUTH
General Authenticate keyref '04'
Decode SW1
SW1='6A'
No
SW1='61'
Get Response
SM-AUTH Verified
Yes
CVC PDVAL
Decode Card UUID from Card CVC
Yes
SW1='90'
No
No
OCC-AUTH
Read buffer 0x1016
Is not empty and does not contain '7F 61 03 02 01 00'?
No
OCC-AUTH Failed
Yes
Decode Tag 0x02
Decode Tag 0x7F60
Livescan fingerprint and prepare template
Verify keyref '96'
If Success
Yes
No
If Tag 0x02 indicates two fingers
Yes
Decode Tag 0x7F60 for second finger
Verify keyref '97'
If Success
Yes
OCC-AUTH Verified
No
No

**Figure 1 - SM/OCC-AUTH checklist**

## 3.2 SM/OCC-AUTH Checklist

As an aid to reading the following checklist, Figure 2 is provided to emphasize the use of two secure channels. One between the card and the validation engine, the second between the reader and the validation engine. The steps shown in the figure use the same numbering found in the checklist below.

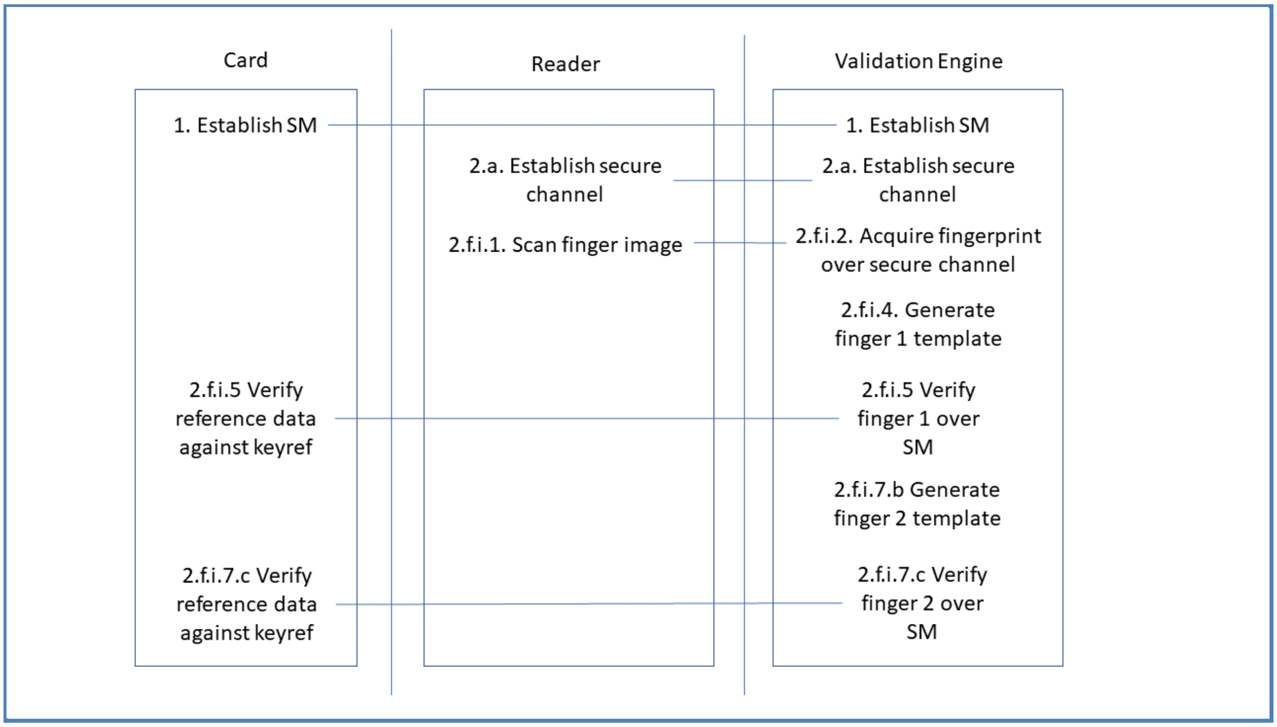

Card
Reader
Validation Engine
1. Establish SM
1. Establish SM
2.a. Establish secure channel
2.a. Establish secure channel
2.f.i.1. Scan finger image
2.f.i.2. Acquire fingerprint over secure channel
2.f.i.4. Generate finger 1 template
2.f.i.5 Verify reference data against keyref
2.f.i.5 Verify finger 1 over SM
2.f.i.7.b Generate finger 2 template
2.f.i.7.c Verify reference data against keyref
2.f.i.7.c Verify finger 2 over SM

**Figure 2 - Secure Channels and checklist command flow**

1. Establish SM between card and validation engine
   a. Select AID PIV
      i. Decode response containing PIV Application Property Template (Tag '61')
      ii. Verify Tag 'AC' is present
      iii. Tag 'AC' will present one or more tags '80' that define cryptographic algorithms supported. There should be only one tag '80' containing a cipher suite for PIV.
         1. Confirm there is a Tag '80' specifying either PIV cipher suite '27' or '2E' (cannot be both)
            a) If yes, this card is capable of Secure Messaging
            b) If no, SM and OCC-AUTH are not supported
   b. Read PIV Secure Messaging Certificate Signer (buffer 0x1017)
      i. If buffer 0x1017 is empty, SM and OCC-AUTH are not supported
      ii. If buffer 0x1017 contains data, SM may be implemented
         1. Extract Secure Messaging Certificate Signer (tag 0x70) using compression information in tag 0x71
            a) If tag 0x70 is empty, SM and OCC-AUTH are not supported

        2. Extract Intermediate CVC if present (tag 0x7F21)[1]
    c. Establish SM crypto channel
      i. If General Authenticate provides SW1=’6A’, then SM and OCC-AUTH are not supported
      ii. If General Authenticate provides SW1=’61’, then there is more to read (i.e. the Card CVC) before the command is finished (APDU Get Response).
      iii. If General Authenticate provides SW1=’90, SW2=’00’, SM is established.
      iv. Parse the Card CVC and extract the Card UUID for identification to the User Record.
      v. If certificate signatures, chaining, and PDVAL on the Secure Messaging Certificate Signer certificate succeeds, SM is supported[2]

2. Validation engine performs OCC-AUTH authentication method
    a. Establish secure channel between validation engine and reader[3]
    b. For contactless, establish SM (steps 1.a.-1.c. above)
    c. Read over secure messaging Biometric Information Templates Group Template (BITGT, buffer 0x1016)[4]
    d. If 0x1016 is empty, or it contains '7F 61 03 02 01 00', OCC-AUTH is not supported
    e. If 0x1016 is not empty and it does not contain '7F 61 03 02 01 00'
    f. Validation engine performs the following:
      i. Decode tag 0x02 to determine number of fingers available
        1. Reader performs live scan of finger
        2. Acquire fingerprint image from reader
        3. Decode tag 0x7F60 for first finger
        4. Prepare template from image for first finger
        5. Verify APDU over secure messaging against key reference ‘96’ for the primary finger
        6. If success, OCC-AUTH verified
        7. If Verify fails
          a) Decode optional tag 0x7F60 for second finger (if present) and prepare live scan
          b) Prepare template from image for second finger
          c) Verify APDU over secure messaging against key reference ‘97’ for the second finger
          d) If success, OCC-AUTH verified

[1] Neither the Card CVC nor the Intermediate CVC contain a CRL Distribution Point (cDP) nor an Authority Information Access (AIA) for revocation/PDVAL status checking. The signature on these certificates shall be valid, and they shall properly chain to the Secure Messaging Certificate Signer.

[2] PDVAL for the Secure Messaging Certificate Signer may be cached by the head-end/validation system. This requires “time of registration” software to read 0x1017 (or get if from a trusted source) and cache the contentSigner. If this certificate is not valid, the card is not valid for access and shall be denied (all authentication methods).

[3] This shall use a mutual authentication channel (PKI) over Encrypted OSDP or TLS between reader and validation engine. This protects the reader to validation engine transmission of the biometric fingerprint images.

[4] This shall be done over an SM encrypted tunnel to enforce the integrity of the object. This mitigates the requirement for the validation engine to read the CHUID and Security Object to confirm the hash of buffer 0x1016 prior to its use.

# 4. Specification Requirements

## 4.1 Determining if Secure Messaging (SM) is supported

The PIV Card Application Property Template is typically coded by the manufacturer of the card. Depending on the card manufacturer's product and the issuer's CMS, it can be overwritten by the issuer.  It states the capabilities of the card and is a necessary condition if a PIV card supports SM.

### 4.1.1 Decode Application Property Template (APT)

The first step is to "Select AID" for the PIV Applet.  Part of the response is Tag '61', the PIV Card Application Property Template.  This is used to determine if the manufacturer (and issuer) enabled SM within its PIV applet.

1. PIV Card Application Property Template is Tag '61' within the Select AID response
   a. The contents of Tag '61' are shown in Figure 3.

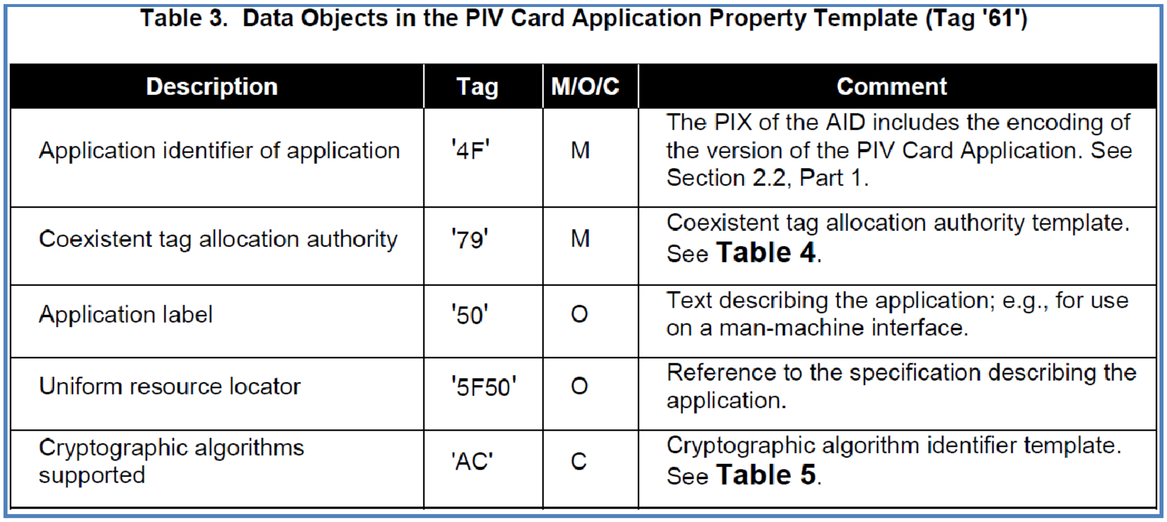


**Table 3.  Data Objects in the PIV Card Application Property Template (Tag '61')**

| Description | Tag | M/O/C | Comment |
|---|---|---|---|
| Application identifier of application | '4F' | M | The PIX of the AID includes the encoding of the version of the PIV Card Application. See Section 2.2, Part 1. |
| Coexistent tag allocation authority | '79' | M | Coexistent tag allocation authority template. See **Table 4**. |
| Application label | '50' | O | Text describing the application; e.g., for use on a man-machine interface. |
| Uniform resource locator | '5F50' | O | Reference to the specification describing the application. |
| Cryptographic algorithms supported | 'AC' | C | Cryptographic algorithm identifier template. See **Table 5**. |

**Figure 3 - [SP 800-73] Part 2, PIV Card Application  Property Template**

   b. Tag 'AC' may contain a series of '0x80' tags defining cryptographic algorithms and one PIV cipher suite supported by the PIV card.  The cipher suites are defined in Figure 4.

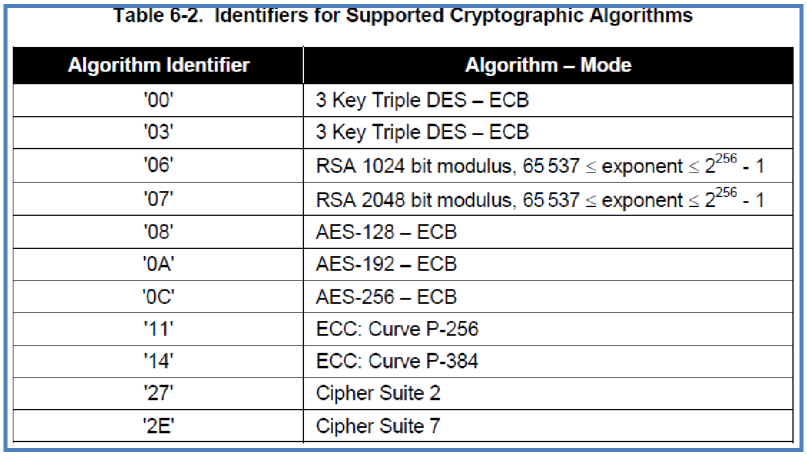


Table 6-2. Identifiers for Supported Cryptographic Algorithms

| Algorithm Identifier | Algorithm – Mode |
|---|---|
| '00' | 3 Key Triple DES – ECB |
| '03' | 3 Key Triple DES – ECB |
| '06' | RSA 1024 bit modulus, 65 537 ≤ exponent ≤ $2^{256}$ - 1 |
| '07' | RSA 2048 bit modulus, 65 537 ≤ exponent ≤ $2^{256}$ - 1 |
| '08' | AES-128 – ECB |
| '0A' | AES-192 – ECB |
| '0C' | AES-256 – ECB |
| '11' | ECC: Curve P-256 |
| '14' | ECC: Curve P-384 |
| '27' | Cipher Suite 2 |
| '2E' | Cipher Suite 7 |

**Figure 4 - [SP800-78] Table 6-2, Supported Cryptographic Algorithms**

c. If ‘27’ is present, then Cipher Suite 2 is supported.
   Cipher Suite 2 = Secure Messaging is using P-256.
d. If ‘2E’ is present, then Cipher Suite 7 is supported.
   Cipher Suite 7 = Secure Messaging is using P-384.

[SP800-78] Section 6.2, Algorithm identifiers '27' and '2E' represent suites of algorithms and key sizes for use with secure messaging and key establishment. Cipher Suite 2 (CS2) is the cipher suite used to establish session keys and for secure messaging when the PIV Secure Messaging key is an ECDH (Curve P-256) key, and Cipher Suite 7 (CS7) is the cipher suite used to establish session keys and for secure messaging when the PIV Secure Messaging key is an ECDH (Curve P-384) key.

When Cipher Suite 2 or 7 are supported, only one is allowed.  Both are not allowed on the same card, per the following:

[SP800-73] Part 2, Section 3.1.1. The presence of algorithm identifier '27' or '2E' indicates that the corresponding cipher suite is supported by the PIV Card Application for secure messaging and that the PIV Card Application possesses a PIV Secure Messaging key of the appropriate size for the specified cipher suite. Tag 0xAC shall be present and indicate algorithm identifier 0x27 or 0x2E (but not both) when the PIV Card Application supports secure messaging.

When one of 0x27 or 0x2E are present, this text states “…the PIV Card Application supports secure messaging.”  This is an incomplete view.  When the APT says SM is supported by the card, it means the card manufacturer enabled SM on the PIV card platform and the PIV applet. It does not indicate that the issuer enabled Secure Messaging on the PIV card that is responding to the reader/application.

This must be verified by determining if the SM CVC certificates and keys are present as established by the issuer.

## 4.1.2 Secure Messaging Certificate Signer

The second step is to verify the Secure Messaging Content Signing certificate(s) are present and valid.

If SM is encoded for a given PIV card, these certificates shall be present and properly coded, providing the trust chain to verify the SM Card CVC.  If container 0x1017 is not present, SM is not enabled on a given PIV card.

The application must Get Data for PIV card data model buffer 0x1017 and decode the information.

This buffer is detailed in Figure 5 and Figure 6.  The CertInfo byte, detailed in Figure 7, is used to determine if a given certificate is compressed or not.  In this instance, the CertInfo byte references tab 0x70, the Content Signing cert, not 0x7F21, the Intermediate CVC.  As stated in Figure 5, the Intermediate CVC shall be stored in uncompressed form.

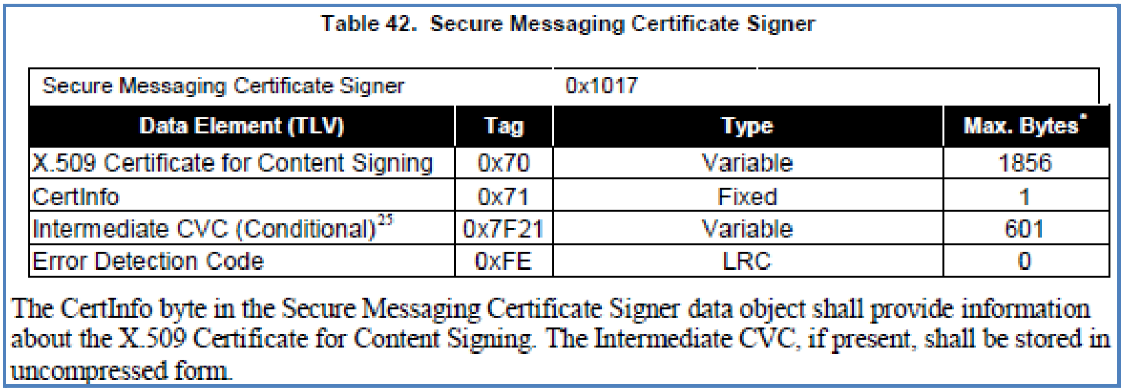
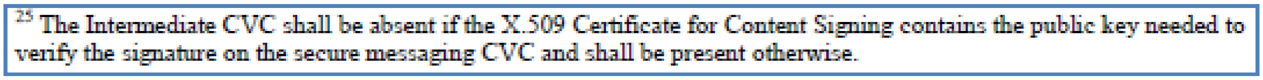


**Table 42.  Secure Messaging Certificate Signer**

| Secure Messaging Certificate Signer | | 0x1017 | |
|---|---|---|---|
| **Data Element (TLV)** | **Tag** | **Type** | **Max. Bytes*** |
| X.509 Certificate for Content Signing | 0x70 | Variable | 1856 |
| CertInfo | 0x71 | Fixed | 1 |
| Intermediate CVC (Conditional)[25] | 0x7F21 | Variable | 601 |
| Error Detection Code | 0xFE | LRC | 0 |

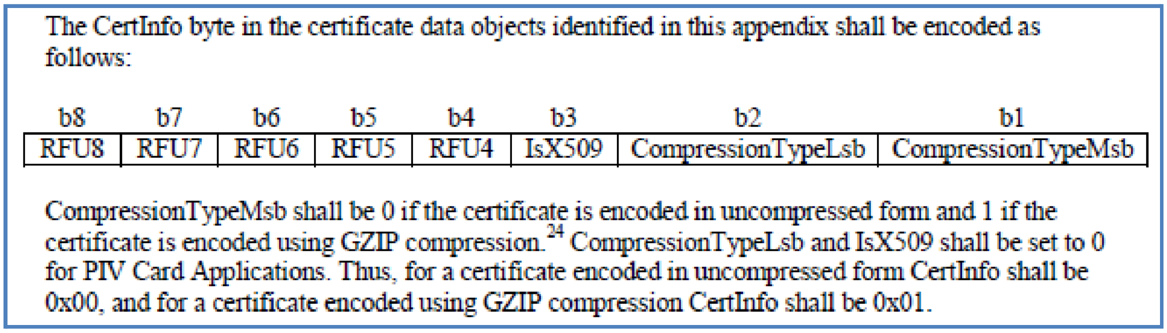
The CertInfo byte in the Secure Messaging Certificate Signer data object shall provide information about the X.509 Certificate for Content Signing. The Intermediate CVC, if present, shall be stored in uncompressed form.

**Figure 5 - [SP800-73] Part 1, Appendix A, SM Certificate Signer buffer**



[25] The Intermediate CVC shall be absent if the X.509 Certificate for Content Signing contains the public key needed to verify the signature on the secure messaging CVC and shall be present otherwise.

**Figure 6 - [SP800-73] Part 1, Appendix A, Footnote 25**



The CertInfo byte in the certificate data objects identified in this appendix shall be encoded as follows:

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|
| RFU8 | RFU7 | RFU6 | RFU5 | RFU4 | IsX509 | CompressionTypeLsb | CompressionTypeMsb |

CompressionTypeMsb shall be 0 if the certificate is encoded in uncompressed form and 1 if the certificate is encoded using GZIP compression.[24] CompressionTypeLsb and IsX509 shall be set to 0 for PIV Card Applications. Thus, for a certificate encoded in uncompressed form CertInfo shall be 0x00, and for a certificate encoded using GZIP compression CertInfo shall be 0x01.

**Figure 7 - [SP800-73] Part 1, Appendix A,  CertInfo byte**

### 4.1.3 Establish Secure Messaging

The third step is to attempt to use secure messaging between the validation engine and the card. There are two expected results:

1. The issuer encoded SM keys and the protocol responds with the Card CVC.
2. The issuer did not encode SM keys, the protocol fails, and the Card CVC is not returned.

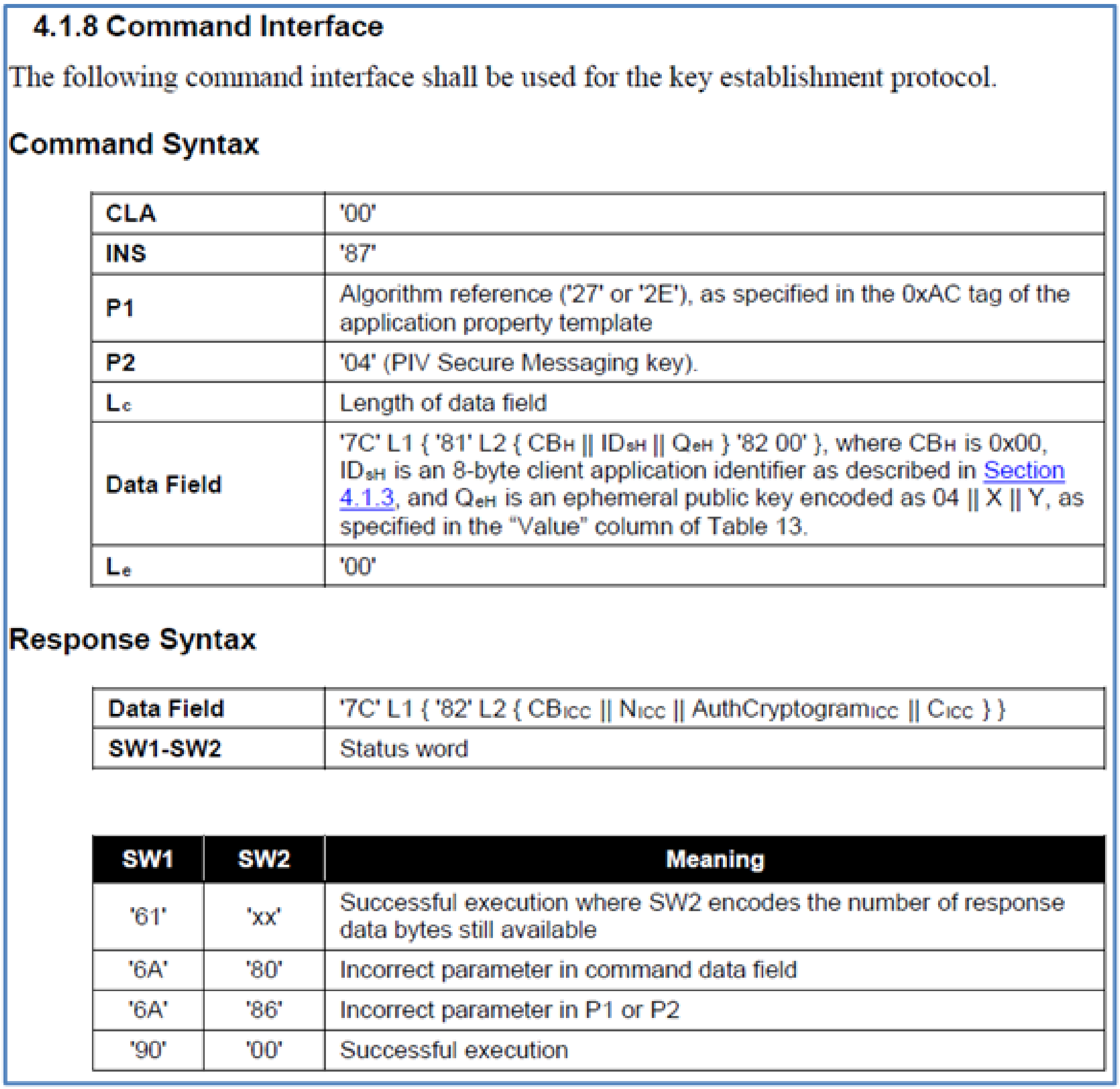
This is detailed in Figure 8. If the issuer encoded SM keys, the card will respond with:

1. SW1=’61’, SW2=’xx’
   General Authenticate command succeeded, but there is more data to be read. SW1=’61’ should be followed with a Get Response command until SW1=’90’, SW2=’00’ is returned telling the caller all data has been returned from the card to the calling application. SW1=’61’ is typical if the CVC data exceeds 256 bytes.
2. SW1=’90, SW2=’00’
   General Authenticate command succeeded, and there is no more data to be read.

If the issuer did not encode SM keys, or an incorrect parameter was sent by General Authenticate (e.g., wrong cipher suite), the card will respond with SW1=’6A’. Essentially this means the card could not figure out what key was asked for, because the requested key does not exist on the card.

If the issuer encoded SM keys, and the General Authenticate command returned SW1=’90, SW2=’00’, then the SM cryptographic protocol is properly initiated. The ePACS must complete Section 4.1.4 below, before responding that SM is established.

Upon successful SM establishment, the Card CVC shall be decoded to extract the Card UUID for identification.



### 4.1.8 Command Interface

The following command interface shall be used for the key establishment protocol.

**Command Syntax**

| | |
|---|---|
| **CLA** | '00' |
| **INS** | '87' |
| **P1** | Algorithm reference ('27' or '2E'), as specified in the 0xAC tag of the application property template |
| **P2** | '04' (PIV Secure Messaging key). |
| **$L_c$** | Length of data field |
| **Data Field** | '7C' L1 { '81' L2 { $CB_H$ \|\| $ID_{sH}$ \|\| $Q_{eH}$ } '82 00' }, where $CB_H$ is 0x00, $ID_{sH}$ is an 8-byte client application identifier as described in Section 4.1.3, and $Q_{eH}$ is an ephemeral public key encoded as 04 \|\| X \|\| Y, as specified in the "Value" column of Table 13. |
| **$L_e$** | '00' |

**Response Syntax**

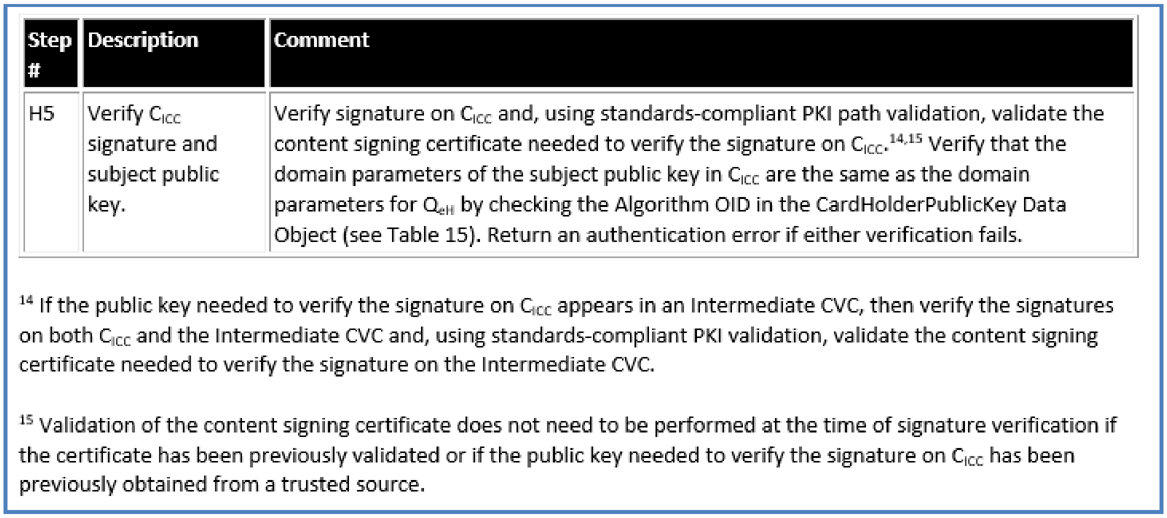
| | |
|---|---|
| **Data Field** | '7C' L1 { '82' L2 { $CB_{ICC}$ \|\| $N_{ICC}$ \|\| AuthCryptogram$_{ICC}$ \|\| $C_{ICC}$ } } |
| **SW1-SW2** | Status word |

| SW1 | SW2 | Meaning |
|---|---|---|
| '61' | 'xx' | Successful execution where SW2 encodes the number of response data bytes still available |
| '6A' | '80' | Incorrect parameter in command data field |
| '6A' | '86' | Incorrect parameter in P1 or P2 |
| '90' | '00' | Successful execution |

**Figure 8 - [SP800-73] Part 2, Section 4.1.8**

## 4.1.4 Secure Messaging CVC PDVAL

[SP800-73] Part 2 requires validation of the certificate chain, from the Card CVC up to a trust anchor (see Figure 9).



| Step # | Description | Comment |
|---|---|---|
| H5 | Verify $C_{ICC}$ signature and subject public key. | Verify signature on $C_{ICC}$ and, using standards-compliant PKI path validation, validate the content signing certificate needed to verify the signature on $C_{ICC}$.[14,15] Verify that the domain parameters of the subject public key in $C_{ICC}$ are the same as the domain parameters for $Q_{eH}$ by checking the Algorithm OID in the CardHolderPublicKey Data Object (see Table 15). Return an authentication error if either verification fails. |

[14] If the public key needed to verify the signature on $C_{ICC}$ appears in an Intermediate CVC, then verify the signatures on both $C_{ICC}$ and the Intermediate CVC and, using standards-compliant PKI validation, validate the content signing certificate needed to verify the signature on the Intermediate CVC.

[15] Validation of the content signing certificate does not need to be performed at the time of signature verification if the certificate has been previously validated or if the public key needed to verify the signature on $C_{ICC}$ has been previously obtained from a trusted source.

**Figure 9 - [SP 800-73] Part 2, Section 4.1.1, Secure Messaging Protocol**

In accord with the protocol, CVC signatures shall be verified. The “when” for “standards-compliant PKI path validation” could significantly impact performance of establishing SM for any given transaction. It is anticipated that an ePACS, at time of registration, will cache the Secure Messaging Signer Certificate for a given issuer’s credentials (note that over time, an issuer will establish more than one contentSigner). Once cached, PDVAL shall be run in accord with APL requirements (every six hours, full trust path validation, full policy OID enforcement).

In the event a Secure Messaging Signer Certificate is no longer valid, as with the CHUID contentSigning certificate, the issuer is deemed compromised and the card is no longer valid for access.

## 4.2 Determining if OCC-AUTH is supported

As shown earlier, the BITGT container is Always available over contact/contactless interfaces. Figure 10 specifies the contents of this container.

OCC is always available over both the contact and contactless interfaces. [SP 800-73] Table 4a, places a security condition on using OCC over contactless, requiring Secure Messaging be in place before OCC.

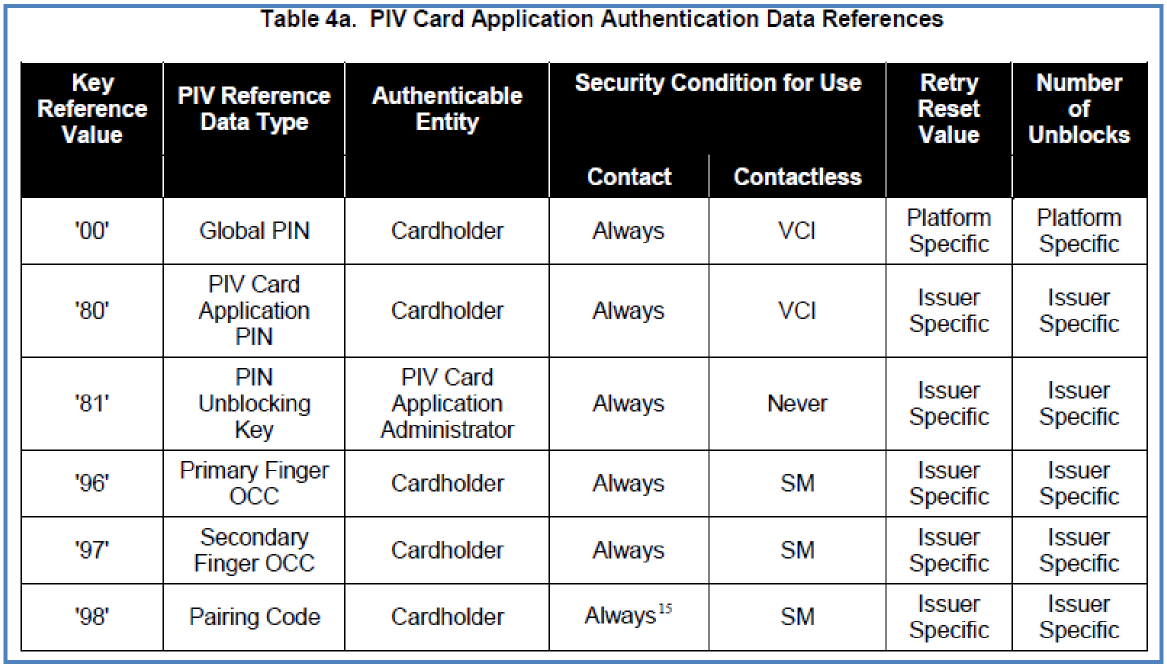


Table 4a. PIV Card Application Authentication Data References

| Key Reference Value | PIV Reference Data Type | Authenticable Entity | Security Condition for Use | | Retry Reset Value | Number of Unblocks |
|---|---|---|---|---|---|---|
| | | | Contact | Contactless | | |
| '00' | Global PIN | Cardholder | Always | VCI | Platform Specific | Platform Specific |
| '80' | PIV Card Application PIN | Cardholder | Always | VCI | Issuer Specific | Issuer Specific |
| '81' | PIN Unblocking Key | PIV Card Application Administrator | Always | Never | Issuer Specific | Issuer Specific |
| '96' | Primary Finger OCC | Cardholder | Always | SM | Issuer Specific | Issuer Specific |
| '97' | Secondary Finger OCC | Cardholder | Always | SM | Issuer Specific | Issuer Specific |
| '98' | Pairing Code | Cardholder | Always[15] | SM | Issuer Specific | Issuer Specific |

**Figure 10 - [SP800-73] Part 1, Table 4a, Security Conditions for use of OCC**

OCC-AUTH is defined such that SM shall be established first, then OCC Verify is performed, which is consistent with [SP800-73] Part 1, Table 4a. Once SM is established, the next steps are to:

1. Establish a secure channel between the reader and the validation engine[5]
2. Try OCC over SM.

If the BITGT container (Figure 11) ID 0x1016 is present and there is data, OCC is likely supported. Read buffer 0x1016 over SM. The BITGT must be decoded to determine how to interact with the card for successful OCC. The BITGT data object encodes the configuration information of the OCC data.

There are two conditions where OCC is not supported by a PIV card. First, BITGT data object is not encoded on the card. Second, it is encoded but specifies there are no BITs within the object as specified in [SP800-73] Part 1, Section 3.3.6, footnote 9:

> [9] A BIT Group Template with no BITs is encoded as '7F 61 03 02 01 00'.
> If either of these conditions are not true, proceed with decoding the object.

Each tag 0x7F60 is called a Biometric Information Template (BIT). [SP800-73] Part 1, Section 3.3.6, provides the rules determining if OCC is present and encoded on the card:

> When OCC satisfies the PIV ACRs for PIV data objects access
> and command execution both the Discovery Object and the BIT

[5] Using digital certificates and either Encrypted OSDP or TLS.

> Group Template data object shall be present, and bit 5 of the first byte of the PIN Usage Policy shall be set.

From a relying party perspective, it is only the BIT Group Template that must be present with well formed BITs to support OCC-AUTH. It is not necessary to read the Discovery Object in order to perform OCC-AUTH. The following sentence is the guide for issuers (not relying parties) when determining how to support OCC-AUTH on their cards:

> The BIT Group Template may be present when OCC does not satisfy the PIV ACRs for PIV data objects access, but, if present, shall contain no BITs.

If OCC does not satisfy PIV ACRs and the BIT Group Template is present (meaning something is in the container), for a fully [SP800-73] compliant card, it shall contain no BITs. This means that OCC will not function, as you need the BITs to determine how to interact with OCC.

The ePACS can interact with the card for OCC-AUTH if the BIT Group Template contains well formed BITs. If the BITs are present, the card supports OCC-AUTH.

The ePACS must decode BIT Group Template Tag 0x02 to determine how many fingers were encoded and attempt OCC. Perform livescan on finger and use the reader to validation secure channel to send the fingerprint image to the validation engine. Prepare template for OCC. Primary finger OCC using the Verify APDU against key reference ‘96’. If successful, OCC-AUTH is verified. If not, and secondary finger is available, secondary finger OCC using the Verify APDU against key reference ‘97’. If successful, OCC-AUTH is verified; if not, OCC-AUTH failed.

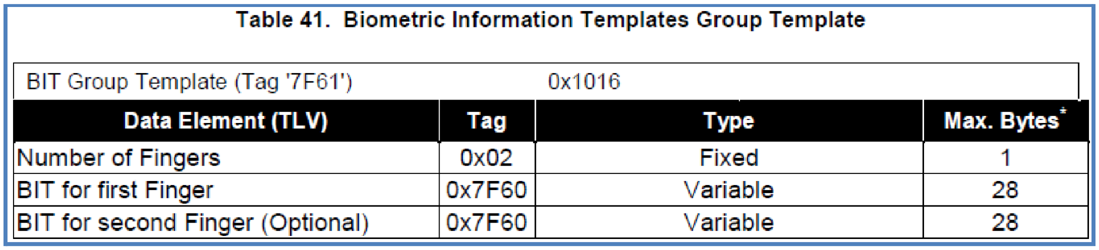


**Table 41. Biometric Information Templates Group Template**

| BIT Group Template (Tag '7F61') | | 0x1016 | |
|---|---|---|---|
| **Data Element (TLV)** | **Tag** | **Type** | **Max. Bytes*** |
| Number of Fingers | 0x02 | Fixed | 1 |
| BIT for first Finger | 0x7F60 | Variable | 28 |
| BIT for second Finger (Optional) | 0x7F60 | Variable | 28 |

**Figure 11 - [SP800-73] Part 2, Table 41**

# 5. Test Cases and Changes

Test Cases added or modified for Secure Messaging and On Card Comparison are added as part of FRTC Version 1.4.2 and are indicated as such within the latest version of the FRTC. Existing positive test cases for registration and time of access were modified to accommodate cards that now contain the elements for SM and OCC. If a system does not use SM or OCC capabilities it should still be able to properly register and grant access to a card that does support SM and OCC.

Systems that choose to use SM and OCC have new time of access test cases created in section 5.19

## 5.1 Secure Messaging and On Card Comparison Test Cases

| Classification | Test Case | Description/Test Case Procedure | Expected Result |
|---|---|---|---|
| SR-1 | 2.16.18 | Various valid PIV cards can be individually registered.<br>SM-AUTH present<br>OCC-AUTH present<br>Trust anchor set to Common Policy Test Root, and the following policies are required:<br>__id-fpki-common-authentication (2.16.840.1.101.3.2.1.48.11)<br>__id-fpki-common-cardAuth (2.16.840.1.101.3.2.1.48.13)<br>__id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) | Registration succeeds. |
| SR-1 | 2.16.19 | Various valid CAC cards can be individually registered.<br>SM-AUTH present<br>OCC-AUTH present<br>Trust anchor set to Common Policy Root, and the following policies are required:<br>__id-fpki-common-authentication (2.16.840.1.101.3.2.1.3.13)<br>__id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.3.39)<br>__id-fpki-common-cardAuth (2.16.840.1.101.3.2.1.3.17) (When applicable) | Registration succeeds. |

| | | | |
|---|---|---|---|
| SR-1 | 2.16.20 | Various valid PIV-I cards can be individually registered.<br>SM-AUTH present<br>OCC-AUTH present<br>Trust anchor set to Common Policy Test Bridge, and the following policies are required:<br>__id-fpki-common-pivi-authentication (2.16.840.1.101.3.2.1.48.78)<br>__id-fpki-certpcy-pivi-cardAuth (2.16.840.1.101.3.2.1.48.79)<br>__id-fpki-certpcy-pivi-contentSigning (2.16.840.1.101.3.2.1.48.80) | Registration succeeds. |
| SR-1 | 5.15.15 | Various valid PIV cards support use of PKI-CAK, PKI-AUTH, PKI-CAK+BIO, PKI-AUTH+BIO, SM-AUTH (Cipher suite '27' and '2E'), OCC-AUTH.<br>Trust anchor set to Common Policy Test Root, and the following policies are required:<br>__id-fpki-common-authentication (2.16.840.1.101.3.2.1.48.11)<br>__id-fpki-common-cardAuth (2.16.840.1.101.3.2.1.48.13)<br>__id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) | Access granted. |
| SR-1 | 5.15.16 | Various valid CAC cards support use of PKI-CAK, PKI-AUTH, PKI-CAK+BIO, PKI-AUTH+BIO, SM-AUTH (Cipher suite '27' and '2E'), OCC-AUTH.<br>Trust anchor set to Common Policy Root, and the following policies are required:<br>__id-fpki-common-authentication (2.16.840.1.101.3.2.1.3.13)<br>__id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.3.39)<br>__id-fpki-common-cardAuth (2.16.840.1.101.3.2.1.3.17) (When applicable) | Access granted. |
| SR-1 | 5.15.17 | Various valid PIV-I cards support use of PKI-CAK, PKI-AUTH, PKI-CAK+BIO, PKI-AUTH+BIO, SM-AUTH (Cipher suite '27' and '2E'), OCC-AUTH.<br>Trust anchor set to Common Policy Test Bridge, and the following policies are required:<br>__id-fpki-common-pivi-authentication (2.16.840.1.101.3.2.1.48.78)<br>__id-fpki-certpcy-pivi-cardAuth | Access granted. |

| | | | |
|---|---|---|---|
| | | (2.16.840.1.101.3.2.1.48.79)<br>__id-fpki-certpcy-pivi-contentSigning<br>(2.16.840.1.101.3.2.1.48.80) | |
| SR-1 | 5.19.1 | Reader set to SM-AUTH mode.<br>Verify Product's ability to verify SM-AUTH using ciphersuite '27'. The explicit policy for Secure Messaging Content Signer contentSigning certificate will be set to the CITE test OID for id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) by the relying party solution. | Access granted. |
| SR-1 | 5.19.10 | Reader set to SM-AUTH mode.<br>Verify product’s ability to reject a credential when the CVC's issuer identification number does not match the subjectKeyIdentifier in the contentSigning certificate when no intermediate CVC is present. | Access denied. |
| SR-1 | 5.19.11 | Reader set to SM-AUTH mode.<br>Verify product’s ability to reject a credential when the Intermediate CVC's issuer identification number does not match the subjectKeyIdentifier in the contentSigning certificate. | Access denied. |
| SR-1 | 5.19.12 | Reader set to SM-AUTH mode.<br>Verify product’s ability to reject a credential when the CVC's issuer identification number does not match the subjectKeyIdentifier in the intermediate CVC. | Access denied. |
| SR-1 | 5.19.13 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when Secure Messaging Content Signer contentSigning certificate is expired. The explicit policy for Secure Messaging Content Signer contentSigning certificate will be set to the CITE test OID for id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) by the relying party solution. | Access denied. |
| SR-1 | 5.19.14 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when Secure Messaging Content Signer contentSigning certificate is revoked. The explicit policy for Secure Messaging Content Signer contentSigning certificate will be set to the CITE test OID for id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) by the relying party solution. | Access denied. |
| SR-1 | 5.19.15 | Reader set to SM-AUTH mode. | Access |

| | | | |
|---|---|---|---|
| | | Verify product’s ability to recognize when the Secure Messaging Certificate Signer contentSigning certificate does not express EKU policy OID 2.16.840.1.101.3.6.7 for content-signers. | denied. |
| SR-1 | 5.19.16 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when Secure Messaging Content Signer contentSigning certificate contains policy '1.2.3.4.5.6'. The explicit policy for Secure Messaging Content Signer contentSigning certificate will be set to the CITE test OID for id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) by the relying party solution. | Access denied. |
| SR-1 | 5.19.17 | Reader set to OCC-AUTH mode (requires SM).<br>Verify product's ability to reject credential when BITGT (buffer 0x1016) is empty. | Access denied. |
| SR-1 | 5.19.18 | Reader set to OCC-AUTH mode (requires SM).<br>Verify product's ability to reject credential when BITGT (buffer 0x1016) contains ‘7F 61 03 02 01 00’. | Access denied. |
| SR-1 | 5.19.19 | Reader set to OCC-AUTH mode (requires SM).<br>Verify Product's ability to accept a OCC fingerprint matching either fingerprint keyref tags '96' or '97'. | Access granted. |
| SR-1 | 5.19.2 | Reader set to SM-AUTH mode.<br>Verify Product's ability to verify SM-AUTH using ciphersuite '2E'. The explicit policy for Secure Messaging Content Signer contentSigning certificate will be set to the CITE test OID for id-fpki-common-piv-contentSigning (2.16.840.1.101.3.2.1.48.86) by the relying party solution. | Access granted. |
| SR-1 | 5.19.20 | Reader set to OCC-AUTH mode (requires SM).<br>Verify Product's ability to reject a valid credential when non-matching OCC fingerprint is presented for fingerprint keyref tags '96' and '97'. | Access denied. |
| SR-1 | 5.19.3 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when PIV Application Property Template Tag '61' does not contain Tag 'AC'. | Access denied. |

| | | | |
|---|---|---|---|
| SR-1 | 5.19.4 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when PIV Application Property Template Tag '61' contains Tag 'AC' but it does not contain a tag '80' reference of '27' or '2E'. | Access denied. |
| SR-1 | 5.19.5 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when buffer 0x1017 is not present. | Access denied. |
| SR-1 | 5.19.6 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when buffer 0x1017 is empty. | Access denied. |
| SR-1 | 5.19.7 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when buffer 0x1017 does not contain tag 0x70. | Access denied. |
| SR-1 | 5.19.8 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when buffer 0x1017 tag 0x70 is empty. | Access denied. |
| SR-1 | 5.19.9 | Reader set to SM-AUTH mode.<br>Verify product's ability to reject credential when Secure Messaging General Authenticate against keyref '04' using ciphersuite '2E' does not result in final SW1='90'. | Access denied. |

# References

| | |
|---|---|
| [FIPS 201] | Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201-2, August 2013, or as amended<br>http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.201-2.pdf |
| [FRTC] | FIPS 201 Evaluation Program Functional Requirements and Test Cases, August 2018<br>https://www.idmanagement.gov/docs/pacsapp-frtcworkbook.xlsx |
| [SP800-73] | Interfaces for Personal Identity Verification - Part 1: PIV Card Application Namespace, Data Model and Representation, NIST Special Publication 800-73-4, May 2015, or as amended<br>https://csrc.nist.gov/publications/detail/sp/800-73/4/final |
| [SP800-78] | Cryptographic Algorithms and Key Sizes for Personal Identity Verification, NIST Special Publication 800-78-4, May 2015, as amended<br>https://csrc.nist.gov/publications/detail/sp/800-78/4/final |