DMDC

Card Technologies & Identity Solutions Division (CTIS)

DoD Implementation Guide for CAC PIV End-Point

CTIS - Technical Advisory Group (TAG)
Version 1.22

07/10/2008

U.S. Department of Defense (DoD)
Defense Manpower Data Center (DMDC)
Card Technologies & Identity Solutions Division (CTIS)
CTIS – Technical Advisory Group (TAG)
Dr. Robert Van Spyk
Marlon A. Guarino
Arnoldo Salas

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background

Homeland Security Presidential Directive-12 [HSPD-12] mandates the implementation of a Federal Information Processing Standard 201 [FIPS 201] Personal Identity Verification (PIV) of Federal Employees and Contractors. The Department of Defense (DoD) Common Access Card (CAC) and DoD Public Key Infrastructure (PKI) programs are being aligned to meet this additional set of requirements. This document takes into account recent updates to FIPS 201, SP800-73-1 and SP800-76.

This document acknowledges that the cryptographic parameters specified by National Institute of Standards and Technology (NIST) SP 800-73-1 will evolve to larger key sizes and hash algorithms according to the time line published in SP800-78. Guidance on physical access is being developed in draft SP 800-116.

Note: For the sake of simplicity the term “CAC Next Generation” or “NG” is also referred to as the “CAC Transitional” or “Transitional” in line with NIST SP800-73-1 part 2 or PIV Transitional.

1.2 Purpose

This Guide specifies technical details for implementing PIV II National Institute of Standards and Technology (NIST) Special Publication (SP) 800-73-1 End-Point requirements in the DoD CAC environment. It covers PIV and DoD mandatory and optional capabilities. It takes advantage of the editorial clarifications in the SP 800-73-2 but otherwise does not include v2.

This Guide emphasizes the delta from the previously published 2007 *DOD Implementation Guide for the CAC Next Generation.* This document specifically deals with the PIV as implemented on the CACv2 with Transitional platform. Middleware specifications are discussed in other documents.

1.3 Audience

This guidance is written for those who provide, acquire, test or develop CAC/PIV applications, middleware or applets for the DoD CAC/PIV End-Point Smart Card program.

1.4 Assertions

- Scope of this document is the delta between the CAC Transitional and PIV End-Point.
- Scope is the End-point interfaces and data model as described in parts 1 and 3 - 7 of SP 800-73-1
- The PIV and CAC applications have clearly defined dependencies.
- CAC is the primary application for DoD
- DoD adds PIV mandatory data model elements to the CAC
- This guide focuses on Java Card implementation
- Optional SP 800-73-1 elements may be mandated for internal use by DoD
- Backward compatibility with existing middleware and card
- DoD middleware has the ability to communicate with CAC, CAC Transitional and PIV End-point
- CAC Credentials will be the primary Credentials
- PIV Transitional and PIV End-Point both surface at the card edge and share the same data

The CAC platform baseline requirements, host application, issuance process, and card usage are only mentioned here when required as context.

2 THE DOD CAC ENVIRONMENT

The PIV End-Point as defined in SP 800-73-1 is added to the existing CAC v2 with the Transitional PIV. Thus, the CAC/PIV End-Point supports both the Transitional and the End-Point solution.

The PIV as implemented on the DoD CAC Transitional is largely separate and distinct from the DoD multi-application CAC. It will evolve at its own pace but in the same environment.

The CAC is the standard identification card for active duty military personnel, selected Reservists, DoD civilian employees, and eligible contractor personnel. In 1999, Congress directed the Secretary of Defense to implement smart card technology within the DoD with the objective of increasing efficiency, security, and readiness. The result has been the creation of the CAC. The baseline functionality of the CAC is to provide:

- Logical access to computer systems.
- Personnel identification
- Physical access to buildings
- PKI for signing, encryption, and non-repudiation.

The CAC PIV End-Point is a multi-application smart card. It serves as a token for PKI identity, email, and encryption certificates. Externally, it contains a linear barcode, two-dimensional barcode, magnetic stripe, color digital photograph, and printed text[1].

CAC End-Point currently supports RSA 1024 and SHA1. According to SP 800-78 the cryptographic parameters will evolve over time to address evolving security requirements (see SP800-78).

2.1 CAC PIV Transitional and CAC PIV End-Point Deltas

Key differences between CAC PIV Transitional and the CAC PIV End-Point are:

- Additional Card Edge APDUs for End-Point:
    - GETDATA, PUTDATA, GENERAL AUTHENTICATE, CHANGE REFERENCE DATA, RESET RETRY COUNTER, and GENERATE ASYMMENTRIC KEY PAIR
- Additional data model elements

DoD PIV Authentication Certificate activation modes.

2.2 Interoperability

The CAC Transitional is interoperable within DoD. The CAC End-Point has an additional applet with PIV End-Point functionality for use within DoD and outside DoD. The Transitional capability is not altered. This architecture allows the CAC PIV End-Point to present both the End-Point and Transitional PIV. Although this capability is transparent for the most part, it has introduced new terms (i.e. DoD Activated PIV Auth. Mode vs. DoD Non-activated PIV Auth. Mode). These are further discussed in subsequent sections.

[1] Card physical characteristics are defined in FIPS 201 Section 4.1

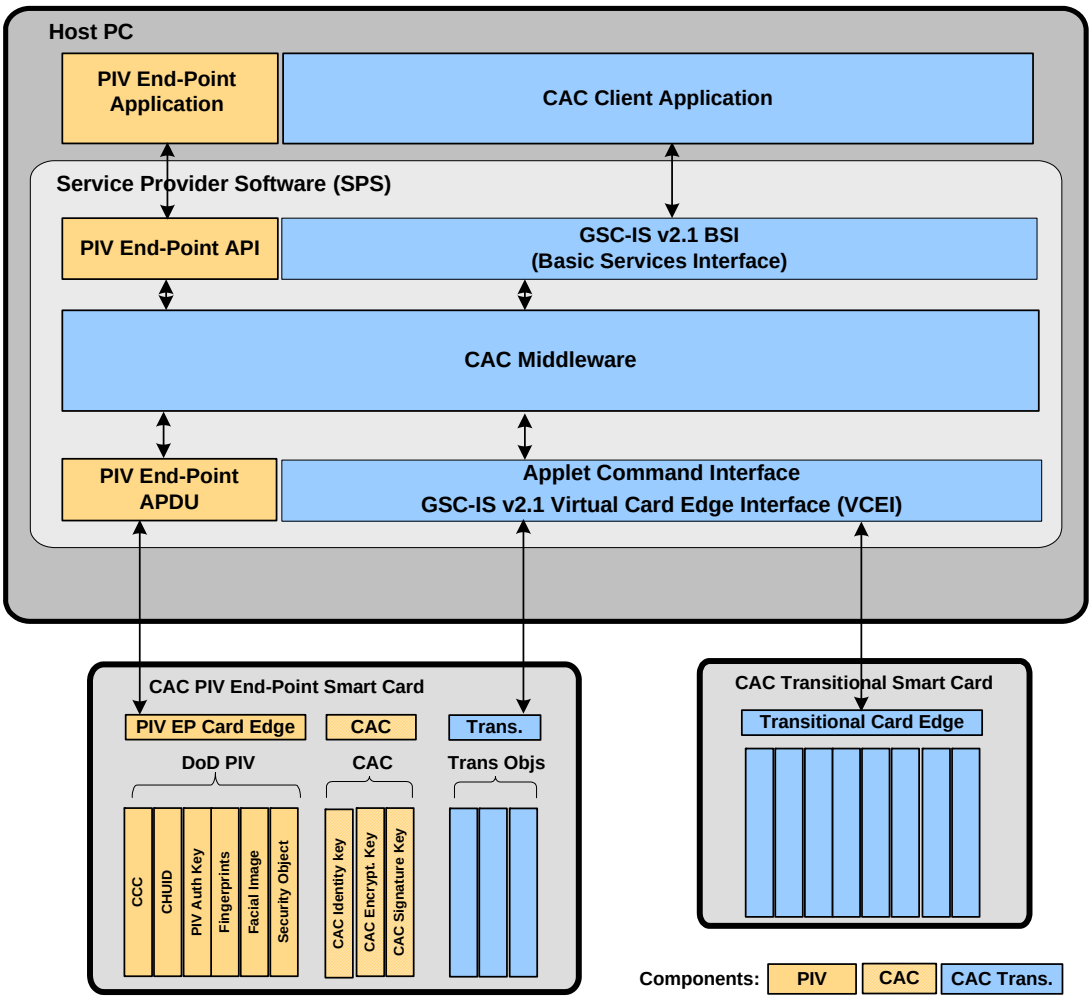
2.3 PIV and CAC Components

Figure 1 below provides a logical representation of the PIV Transitional and End-Point on the CAC platform. The upper square represents a DoD computer hosting CAC or PIV applications and middleware. The two cards beneath it represent an End-Point and a Transitional card.

The right card illustrates the CAC Transitional, which leverages the existing GSC-IS 2.1 [GSC-IS] BSI and card edge to serve existing CAC and CAC Transitional applications.

The left card illustrates the DoD CAC End-Point card. A PIV host application will use the PIV for physical or logical access, communicating via the SP 800-73-1 interfaces in the Transitional and the API for the End-Point.

Figure 1. Sample CAC and PIV components



3 DATA MODEL DISCOVERY

The data model version number was intended to correspond to the scope and version of data objects. However, in current discussions regarding SP 800-73-2 this does not appear to be the case.

Thus, CAC and PIV functionality at the contact interface requires data model discovery. This section presents an example of this discovery process. Discovery is not needed on the contactless interface because the card only surfaces the CHUID.

3.1 Data Model Discovery Combinations

Table 1 summarizes the data models that can be expected.

Table 1. Card Data Model Discovery

Card Type to Discover	CCC (GSCIS-RID or NIST RID for PIV)	Data model ID (0xF5)
CACv1	No CCC implemented.	N/A
CACv2	No CCC implemented.	N/A
CACv2	CCC implemented.	0x02
Contactless Pilot	CCC (GSC-IS RID)	0x02
CAC Transitional	CCC (GSC-IS RID)	0x10
PIV end-point card	CCC exists within PIV Application AID	0x10

3.2 Data Model Discovery for Contact Card

The criteria used to discover the card data model are:

- CCC presence, using GSC-IS 2.1 RID or NIST RID
- Data model version, using CCC data element 0xF5 or application ID (AID)

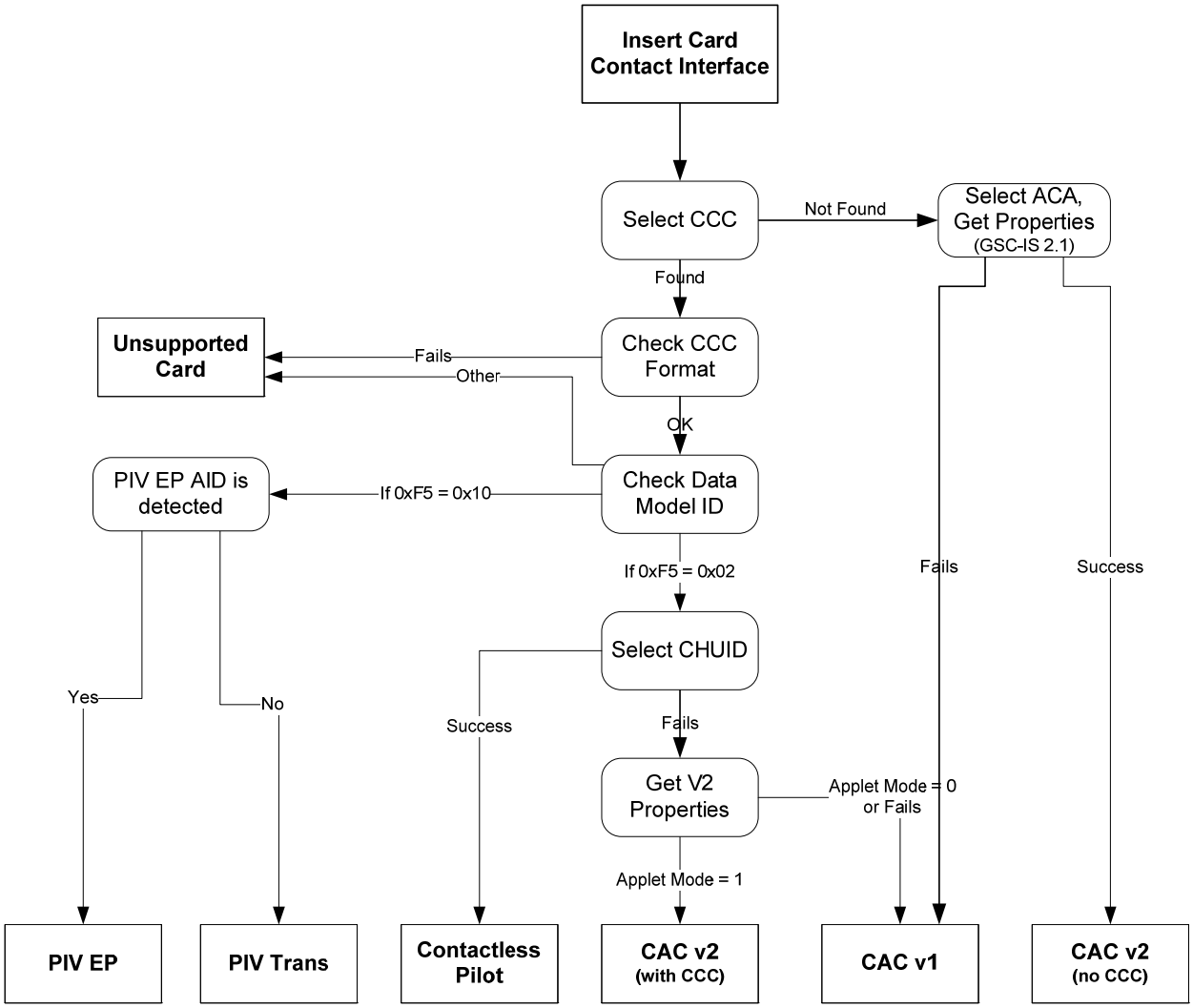
Below is an example of discovery for identifying the data models on a contact card for logical access. Note that the order of discovery steps is important.

Select CCC.
IF CCC not found THEN
IF Select ACA AND get Properties (using GSC-IS 2.1) SUCCESS THEN
RETURN ***CAC V2*** *(no CCC).*
ELSE
RETURN ***CAC V1***
ELSE IF CCC Format = OK THEN
IF data model id tag 0xF5 = 0x02 THEN
Select CHUID
IF SUCCESS THEN
RETURN ***Contactless Pilot***
ELSE Get V2 Properties
IF Applet Mode = 0 OR Fails THEN
RETURN ***CAC V1***
ELSE IF Applet Mode = 1 THEN
RETURN ***CAC V2*** *(with CCC)*
ELSE
RETURN ***Unsupported Card***
END IF ELSE
ELSE IF data model id tag 0xF5 = 0x10 THEN
IF CCC AID = PIV End-Point AID THEN
RETURN ***PIV End-Point****.*
ELSE

RETURN ***PIV Transitional***[2]
END IF ELSE
END IF ELSE
END IF ELSE
RETURN ***Unsupported Card***

Depicted below is a middleware discovery flow diagram for contact card supporting CAC, CAC Transitional, or CAC PIV End-Point.

Figure 2. Contact Card Discovery Flow



3.3 Data Model Discovery for Contactless

Contactless discovery of the PIV Transitional and PIV End-Point is based on the selection of the default container. The default container for both is the CHUID. The minimum interoperability mechanism for cardholder identification is to read the CHUID from a fixed

[2] PIV Transitional is synonymous to CAC NG.

location using READ BINARY and SELECT EF ISO 7816-4 [ISO4]. GET DATA for contactless is an end-point. Physical access device vendors may support both READ BINARY and GET DATA for interoperability.

3.4 Contactless Interoperability

DoD CAC PIV cards support both contact and contactless interfaces.

SP 800-73-1 contactless cards provide a minimum interoperability mechanism for cardholder identification for physical access.

SP 800-73-1 contactless cards and readers shall conform to ISO 14443 Parts 1 through 4. Cryptographic functionality is not required, but GSC contactless cards that implement cryptography use FIPS-approved cryptographic algorithms in FIPS 140-2 validated modules.

3.5 Default Select Container (Object)

The table below summarizes default container depending on card type issued and mode in use.

Table 2. Default Selected Containers

Card Type	Contact	Contactless
CACv2	CardManager	N/A
CAC Transitional	CHUID (Transitional)	CHUID (Transitional)
PIV end-point card	CHUID	CHUID

4 CAC PIV END-POINT DATA MODEL

This section contains the DoD PIV data model. It includes six mandatory containers and four optional containers as mandated by SP 800-73-1. The DoD implements those listed below, including two PIV optional containers which are DoD mandatory. The PIV Transitional and End-Point share most data elements and access control requirements.

DoD implements the following PIV containers:

- CCC
- CHUID
- Security Object
- Card Holder Fingerprints Container (containing primary and secondary fingerprints).
- Facial Image Container (PIV optional but DoD mandatory)
- PIV Auth. Certificate[3]

These optional PIV containers are available through existing CAC keys:[4]

- Digital Signature Key (DoD uses existing CAC PKI Digital Signature key)
- Key Management Key (DoD uses existing CAC PKI Encryption key)

The remaining optional PIV containers are not implemented:

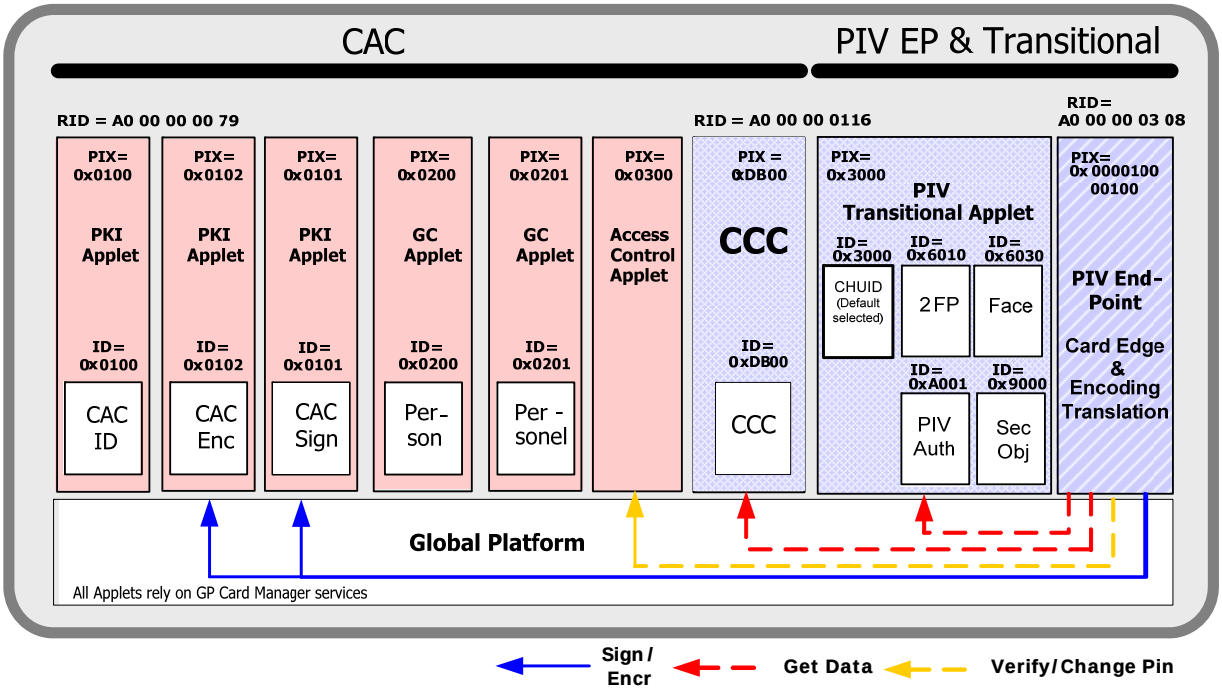
- Card Authentication Key

[3] See Appendix section “Sample data for PIV Authentication Certificate”.

[4] These CAC PKI keys (CAC PKI Digital Signature and CAC PKI Encryption keys) are available to the PIV optional keys (Digital Signature and Key Management key), so that these appear at End-Point card edge. The middleware checks for the required key. The PIV End-Point does not duplicate certificates but rather use existing CAC certificates.

- Printed Information Buffer

Figure 3. CAC PIV End-Point Data Model[5]



SP 800-73-1 permits default applets[6]. In this profile the default selected applet and container for both contact and contactless are the PIV Transitional Applet and the CHUID container. To use PIV defined GET DATA commands after a cold or warm reset a SELECT PIV End-Point Applet command is required. The CCC is shared between PIV End-Point and the Transitional data model.

Note: To access the PIV Auth Certificate from the End-Point card edge, implementers first select the PIV EP applet and then issue a GET_DATA request as per SP800-73. To access the PIV Auth Certificate from the PIV Transitional card edge (PIV Auth Activated mode), implementers use OID 0xA001.

4.1 Access Control Rule Definition

The following table lists the Access Control Rules supported by PIV End-Point applet during the *Card Usage* of the lifecycle phase of card[7].

[5] When using a PIV End-Point card with the PIV Auth activated mode, the PIV Auth Certificate OID at the PIV Transitional card edge will read 0xA001 as opposed to 0x0101.
[6] SP800-73-1 section 3.4.2, pg 19 does not mandate a specific selected application.
[7] These are not applicable at the *Card Issuance* lifecycle phase.

Table 3. Applet Access Control Rules Definitions

ACR Value	ACR Meaning
Always / ALW	The corresponding service can be provided without any restrictions. Always readable without PIN verification.
Never / NEV	The corresponding service can never be provided.
PIN	The corresponding service is provided if the PIN has been successfully validated. PIN must be provided but can be cached for subsequent operations.
PIN Always / PIN ALW	The corresponding service can be provided if the PIV application PIN code value has been verified for each operation.

The following table lists all the BER-TLV tags supported by the PIV End-Point applet, the associated key reference when applicable, as well as the access control rules associated to each of them. Optional objects not implemented by DoD are grayed out.

Table 4. PIV Data Objects Access Control Rules

Data Object	Tag[8]	OID[9]	Key	Contact			Contactless			M / O
For Interoperable Use	BER-TLV		Ref	Read	Upd	Gen Auth	Read	Upd	Gen Auth	
Card Capability Container	5FC107	DB00	N/A	ALW	NEV	N/A	NEV	NEV	N/A	M
Card Holder Unique Identifier	5FC102	3000	N/A	ALW	NEV	N/A	ALW	NEV	N/A	M
X509 Certificate for PIV Authentication	5FC105	A001[10]	9Ah	ALW	NEV	PIN	NEV	NEV	NEV	M
Card Holder Fingerprints	5FC103	6010	N/A	PIN	NEV	N/A	NEV	NEV	N/A	M
Printed Information	5FC109	3001	N/A	PIN	NEV	N/A	NEV	NEV	N/A	O
Card Holder Facial Image	5FC108	6030	N/A	PIN	NEV	N/A	NEV	NEV	N/A	O
X509 Certificate for Digital Signature	5FC10A	0100	9Ch	ALW	NEV	PIN ALW	NEV	NEV	NEV	O
X509 Certificate for Key Management	5FC10B	0102	9Dh	ALW	NEV	PIN	NEV	NEV	NEV	O
X509 Certificate for Card Authentication	5FC101	0500	9Eh	ALW	NEV	ALW	ALW	NEV	ALW	O

[8] These Tag values are used in the GET DATA APDU command, to access individual PIV objects

[9] OID is not interpreted at the applet level, but the link between BER-TLV tags and OIDs is described here for reference only

[10] The PIV Auth certificate OID is 0x0101 if accessed through the End-Point card edge. The 0xA001 OID is only available in the PIV Auth Activated mode of the Transitional card edge.

Security Object	5FC106	9000	N/A	ALW	NEV	N/A	NEV	NEV	N/A	M

Where columns:

- "**Read**" stands for "GET DATA" operations
- "**Upd**" stands for "PUT DATA" operations
- "**Gen Auth**" stands for "GENERAL AUTHENTICATE" operations
- "**M/O**" stands for Mandatory or Optional
- "**N/A**" stands for Not Applicable

4.2 Version Number

A version number is associated with the data model used on the card to aid the host systems determine what is supported. The following hexadecimal version numbers are used for current and future DoD CACs:

- **0x01** – Used for standard GSC-IS profiles. CACv1 has no CCC, hence no data model.
- **0x02** – the Contactless Pilot which is CACv2 plus the CHUID and CCC.
- **0x10** – the CAC Transitional, or CAC End-Point. The middleware must obtain the AID from the CCC to differentiate which one applies.

The data model for CAC may only range from 00-04.

5 DoD PIV End-Point Data Elements

The CAC PIV data model containers are further defined in the following sections.

5.1 CCC (0xDB00)

The CCC is exposed at the contact interface to allow discovery of card capabilities. This is in compliance with SP 800-73-1, and supports minimum capabilities for lookup on data model and application information. The data model number is 0x10.

5.1.1 CCC Usage

The CCC container is used to allow fast and light discovery of card capabilities. There is a single CCC on the card when both PIV EP and CAC applications are present[11]. Middleware implementers should read relevant information from the CCC, rather than hard coding values in the middleware for specific cards.

CAC data objects (keys, containers, etc.) can be addressed directly with a prior knowledge of the generic container AIDs. However, the CCC provides information allowing to:

- Confirm that the data model ID is known, and therefore, the data model scope and format can be handled by the middleware.
- Discover exactly which data objects are present on the card, and where they are located, since GSC-IS provides freedom on how to map data objects onto applet instances. This discovery information located is in the ApplicationsCardURL attributes.

[11] Implementers should be aware that data model number may change as revisions to SP800-73 occur. The current data model number is 0x10.

The table below lists the application URL entries used by the DoD. Current GSC-IS and PIV CCC URL entries are 18 bytes each (tag + length + value). SP 800-73-1 stipulates a maximum of 128 bytes for the variable ApplicationsCardURL data element, which accommodates only 7 total entries. This is fewer than needed to contain all of the entries needed by the DoD. The limit is interpreted as a guideline, not a hard limit.

Table 5. PIV End-Point Application URL Values

RID	App ID	Object ID	Application	Application Card URL
A000000308	3000	3000	Card Holder Unique Identifier (CHUID)	F3 10 A000000308 01 3000 3000 00 00 00000000
A000000308	3000	6010	Card Holder Fingerprints Container	F3 10 A000000308 01 3000 6010 00 00 00000000
A000000308	3000	9000	Security Object	F3 10 A000000308 04 3000 9000 00 00 00000000
A000000308	3000	6030	Facial Image	F3 10 A000000308 01 3000 6030 00 00 00000000
A000000308	3000	0101	PIV Authentication Cert.	F3 10 A000000308 04 A001 0101 00 00 00000000
A000000308	0000	0100	PIV Card Application	F3 10 A000000308 01 0000 0100 00 00 00000000

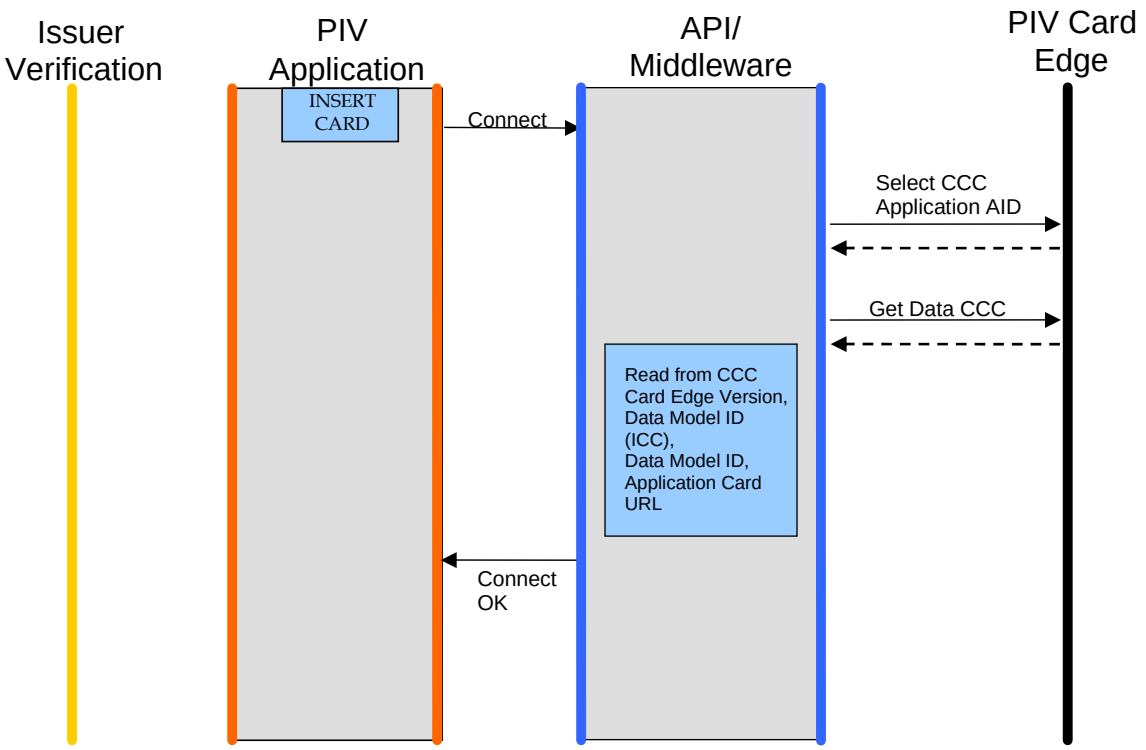
5.1.2 CCC Syntax

The DoD CCC supports minimum capabilities for lookup on data model and application information. The CCC definition in Appendix A of SP 800-73-1 is in line with GSC-IS 2.1 except that the content of the variable length ApplicationsCardURL data element is not defined.

GSC-IS 2.1 defines the components of the CCC ApplicationsCardURL syntax. The length each card URL entry is 16 bytes (plus 2 bytes for tag and length). The corresponding Tag 0xF3 is repeated to list all GSC-IS 2.1 compliant objects on the card. The CCC may be recovered as follows.

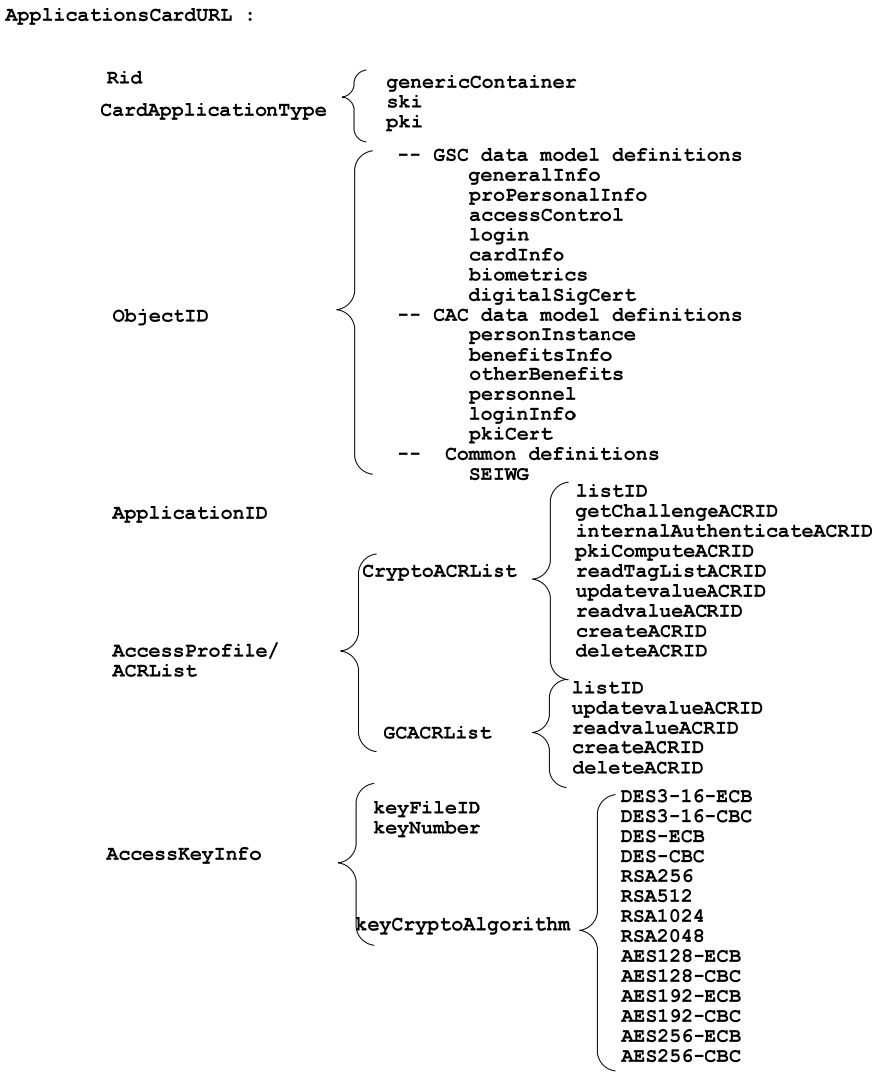
Figure 4. Retrieval of CCC Sequence Diagram

(This process must run before any other transaction may begin)



This section details the format of the PIV CCC, including the ApplicationsCardURL element, combining the specifications from SP 800-73-1 and GSC-IS 2.1. The components and the values allowed in a card URL are depicted in the figure below.

Figure 5. GSC-IS CCC Encoding



5.1.3 CCC CAC v2 and PIV Container Content

SELECT End-Point AID

GET DATA ‘5FC107’ (CCC)

CCC access condition: Read ALW; Update OP-SC

CCC container should contain the following TLV items:

1 / Card Identifier

Tag 0xF0

Len 0x15

Value GSC-RID (5B) | manufacturer-Id (1B) | Card-type (1B) | Card-Id (14B)

(Note: Card-Id = CUID (10B) + Manufacturer batch serial (4B) (ref. DMDC-W2K) CUID & batch serial are defined in CAC re-Issuance Req. v3.9.1a pp. 20-21. An example for GSC-RID is A000000079 Card-type = '02' for Java Card; '01' for File System card.)

2 / Capability Version

Tag 0xF1

Len 0x01

Value 0x21 (meaning GSC-IS v2.1)

3 / Capability Grammar Version

Tag 0xF2

Len 0x01

Value 0x21 (meaning GSC-IS v2.1)

4 / ApplicationsCardURL

Tag 0xF3

Len 0x10

Value RID (5B) e.g. A000000079

AppType (1B) 0x01 = GC; 0x04 = PKI; 0x00 = PIV; 0xA0 for 3rd party applets;

ObjectID (2B) e.g. 0x02FE for PKI certificate.

AppID (2B) See Table 8.

AccProfile not used for Java Card, 00

PinID (1B) not used for Java Card, 00

AccKeyInfo (4B) not used for Java Card, 00000000

See table below for actual values on the DoD PIV CAC

5 / PKCS#15

Tag 0xF4

Len 0x01

Value 0x00 (meaning not PKCS#15 token)

6 / Registered Data Model

Tag 0xF5

Len 0x01

Value 0x10

7 / Access Control Rule Table

Tag 0xF6

Len 0x11

Value 0x07 0xA0 0x00 0x00 0x00 0x79 0x03 0x00 000000000000000000

(CHOICE of acrTableAID)

8 / Card APDUs

Tag 0xF7

Len 0x00 (not used)

9 / Redirection Tag

Tag 0xFA

Len 0x00 (not used)

10 / Capability Tuples

Tag 0xFB

Len 0x00 (not used)

11 / Status Tuples

Tag 0xFC

Len 0x00 (not used)

12 / Next CCC

Tag 0xFD

Len 0x00 (not used)

13 / Error Detection Code

Tag 0xFE

Len 0x00 (not used)

5.2 CHUID (0x3000)

Clarifications of CHUID fields not covered in this document are specified in the "*DoD Implementation Guide for CAC Next Generation (NG)*" [NG]. These include FASC-N, Global Unique Identifier (GUID), Expiration Date, Authentication Key Map, and Error Detection Code.

See Appendix "*Sample Java program: Accessing the CHUID*" for a code sample on retrieving the CHUID.

5.2.1 CHUID Usage

As outlined in the NIST Special Publication 800-73-1 the CHUID is defined by the following table:

Table 6. CHUID Container

Card Holder Unique Identifier		0x3000	Always Read	
Data Element (TLV)	**Tag**	**Type**	**Max Bytes**	**M/O**
FASC-N	0x30	Fixed Text	25	M
GUID	0x34	Fixed Numeric	16	M
Expiration Date	0x35	Date (YYYYMMDD)	8	M
Authentication Key Map	0x3D	Variable	512	O
Issuer Asymmetric Signature	0x3E	Variable	2048	M
Error Detection Code	0xFE	LRC	0	O

- The CHUID includes an element, the Federal Agency Smart Credential Number (FASC-N), which uniquely identifies each card and cardholder.
- The PIV CHUID is a free read from both the contact and contactless interfaces of the PIV Card[12].
- The FASC-N is not allowed to be modified post-issuance.
- In machine readable format, the expiration date data element specifies when the card expires. The expiration date format and encoding rules are as specified in SP800-73-1. This date is the same as that on the printed card surface. The optional Printed Information Buffer is not included.
- Includes the issuer Public Key Certificate in the container.
- The Asymmetric Signature data element of the PIV CHUID has been encoded as a Cryptographic Message Syntax (CMS) external digital signature, as defined in RFC 3852 [RFC3852].
- Algorithm and key size requirements for the asymmetric signature are detailed in [SP800-78].
- Optional fields are not implemented (Grayed out on table above).

[12] Applies to both DoD PIV Auth Activated and Non-activated mode.

The issuer digital signature is computed over the concatenated contents of the CHUID to include the Tag, Length, and Value. The signature excludes the Asymmetric Issuer Signature Field (FIPS 201 4.2.2) and the Authentication Key Map, if present. The Tags are one byte.

5.2.1.1 FASC-N

DoD implementations utilize the first 16 characters of the FASC-N (Agency Code + System Code + Credential Number + Credential Series + Individual Credential Issue) to uniquely identify a PIV card. By using the two additional fields Credential Series and Individual Credential Issue fields as part of the Credential Number increases the limit of 999,999 card issuance per site, to 99,999,999, and it will accommodate larger population.

Table 7. FASC-N

	Field Name	Length (chars.)	Field Description
Uniquely Identifies Card	Agency Code	4	Identifies the government agency issuing the credential
	System Code	4	Identifies the system the card is enrolled in and is unique for each site
	Credential Number	6	Encoded by the issuing agency. For a given system no duplicate numbers are active.
	Credential Series (Series Code)	1	Field is available to reflect major system changes
	Individual Credential Issue	1	Usually a 1 will be incremented if a card is replaced due to loss or damaged
Uniquely Identifies Card Holder	Person Identifier	10	Numeric Code used by the identity source to uniquely identify the token carrier
	Organization Category	1	Type of Organization the individual is affiliated with
	Organization Identifier	4	The Identifier that identifies the organization the individual is affiliated with.
	Person/Organization Association Category	1	Indicates the affiliation type the individual has with the Organization.

The last 16 characters uniquely identify the card holder.

5.2.2 Issuer Asymmetric Signature

The DoD CHUID issuer asymmetric digital signature is applied to all objects on the card. The DoD retrieves the digital certificate from a Defense Information Systems Agency [DISA] approved Certificate Authority. The issuer certificate is different from others on the card in that it does not contain the message digest.

Issuer Asymmetric Signature follows *RFC 3852, Cryptographic Message Syntax*. The issuer asymmetric signature file is implemented as a SignedData Type, as specified in Distinguished Encoding Rule (DER) format to preserve the integrity of the signatures within them.

The ChuidSecurityObject in the CHUID signature MessageDigest attribute is an encrypted hash of all the CHUID elements, except for the asymmetric signature and key map. The key computations are outlined as follows:

- A binary string representing the plain-text concatenated from all of the following DoD mandatory elements
    - FASC-N
    - GUID
    - Expiration Date
- A Message Authentication Code (MAC) is then computed on this plain-text string by the card issuer, using the digestAlgorithm specified in the SignedData object.

The resulting MAC is signed by the card issuer, using the `signatureAlgorithm` specified in `SignedData` object defined below. **Note:** Neither this signature nor the MAC are part of the PIV Security Object container defined later in this section. It is part of the asymmetric signature field's `SignedData` object. The CHUID signature is as follows:

Table 8. Issuer Asymmetric Signature "SignedData" Object

Value		Comments
SignedData		
CMS version	m	Value = v3
digestAlgorithms	m	As specified in SP 800-78.
encapcontentInfo	m	
eContentType	m	id-PIV-CHUIDSecurityObject
eContent	x	This field "shall" be omitted (FIPS 201)
certificates	m	Issuers shall include only a single X.509 certificate, the Document Signer Certificate (C_{DS}), which is used to verify the signature in the SignerInfo field.
crls	x	This field "shall" be omitted (FIPS 201)
signerInfos	m	This field "shall" be present and include a single SignerInfo (FIPS 201)
SignerInfo	m	
CMS version	m	Version must be 1 because of mandated sid choice. (See RFC3852 Section 5.3 for rules regarding this field).
sid	m	Signer Identifier
issuerandSerialNumber	m	This field "shall" use the 'issuerAndSerialNumber' choice (FIPS 201)
digestAlgorithm	m	The algorithm identifier of the algorithm, specified in SP 800-78, used to produce the hash value over SignedAttrs.
signedAttrs	m	Issuers may wish to include additional attributes for inclusion in the signature. However, these do not have to be processed by receivers except to verify the signature value. FIPS 201 and RFC 3852 specify that, at a minimum, the SignerInfo shall include the next three attributes.
ContentType		id-PIV-CHUIDSecurityObject

MessageDigest	m	The hash over the CHUID data as described previously.
pivSigner-DN	m	The subject name that appears in the PKI certificate for the entry that signed the CHUID.
signatureAlgorithm	m	The algorithm identifier of the algorithm used to produce the signature value, and any associated parameters.
signature	m	Encrypted hash of signed attributes that results from the signature generation process.

The following processing rules in RFC3852 apply to the table above:

m (mandatory) the field MUST be present
x (do not use) the field SHOULD NOT be populated
o (optional) the field MAY be present
c (choice) the field contents is a choice from alternatives

5.3 Security Object (0x9000)

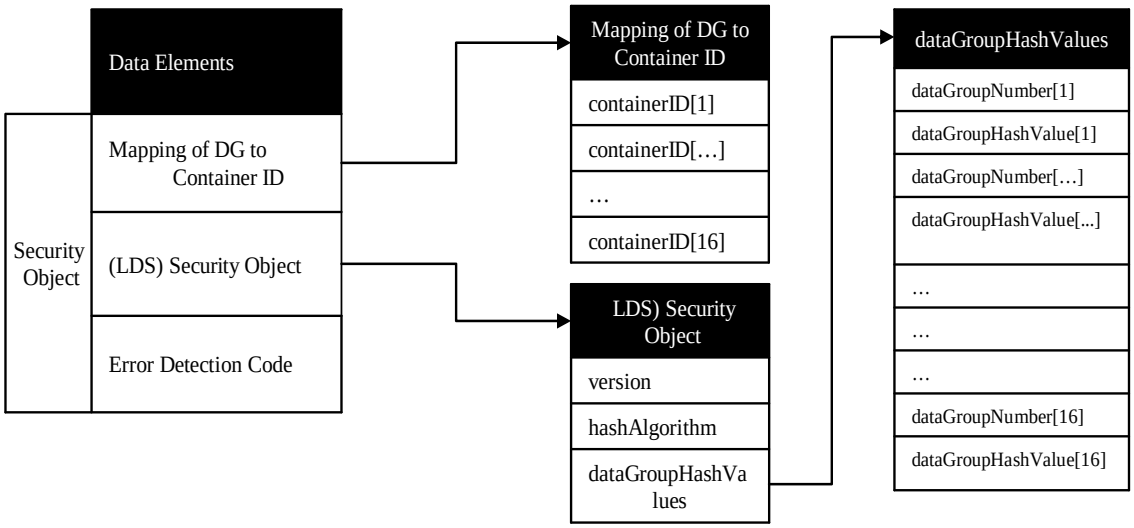
The principal goal of the security object is to reduce the cryptographic operations. It enables the binding of the issuer to various data objects using a single signature, yet does not require all of the data to be read from the card to verify the signature.

The Security Object provides a means for verifying the integrity of card data elements that bind a card to the card holder's identity with minimal processing. It is signed by the issuer. Again, the issuer's certificate is not included with the Security Object, since it is already part of the CHUID.

The Security Object contains hashes of the entire contents of the fingerprint and facial image containers (TLV and including their signature). For containers not present on the card, the container IDs mapped to their respective Data Groups is null.

The card issuer's signature key used to sign the CHUID is also used to sign the Security Object. The signature field of the Security Object, Tag "0xBB" omits the issuer's certificate, since it is included in the CHUID. The Security Object does not contain the three optional unsigned data elements 1) Printed Information data object, 2) Unsigned CHUID, and 3) Facial Image data object is included in the Security Object.

Figure 6. Security Object Structure



Note: In SP800-73-1, the data structure for "Mapping of DG to Container ID" is not defined.

5.3.1 Security Object Specification

The SP 800-73-1 Security Object container includes an element (0xBA) that maps the MRTD data groups to the corresponding PIV data model container ID and a signedData type element (0xBB) that contains hashes of the mapped containers and is signed by the issuer using the card issuer's digital signature key as in the LDS Security Object. The hashes are those in the signature of each signed object. The hashes include the entire contents – even the signature. The hashes are placed in the security object in the order in which data elements are presented in the PIV data model overview. However, the order is not relevant. The issuer's certificate is not included in the Security Object element, since it is already part of the issuer certificate in the CHUID.

The Security Object Container is described in SP 800-73-1 according to the following tables:

Table 9. Security Object Container

Container Description Container	ID	Maximum Length (Bytes)	Access Rule	Contact/ Contactless	Mandatory/ Optional
Security Object	0x9000	1000	Always Read	Contact	Mandatory

Table 10. Security Object Container Elements

Security Object (PIV)	0x9000		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Mapping of DG to container ID	0xBA	Variable	100
LDS Security Object (MRTD Document SO)	0xBB	Variable	900
Error Detection Code	0xFE	LRC	0

5.3.2 Mapping of Data Groups to PIV Containers

DoD implementation uses explicit mapping and fully populated 16 entry table with 1 byte index followed by 2 byte container ID.

The PIV Security Object contains the hash for PIV containers, which, when present on the card, are explicitly specified for mandatory integrity protection by the Security Object in SP 800-73-1.

For informational purposes the following table cross-references the relevant Security Object data elements with the Data Group hash values.

Table 11. Mapping of Data Groups to Container ID

DataGroup Number	container ID	dataGroup HashValue	References to SP800-73-1	Comment
1	null	null		Machine Readable Zone (MRZ)
2	0x6030	MIT	Image for Visual Verification (to be consistent with previous references in this document)	Encoded Face.
3	0x6010	MIT	Card Holder Fingerprints (Appendix A)	Similar to LDS Encode Finger Datagroup [MRTD].
4	null	null		Encoded Iris
5	null	null		Similar to LDS Display Portrait Datagroup [MRTD].
6	null	null		Reserved for future [MRTD]
7	null	null		Displayed Signature [MRTD]
8	null	null		
9	null	null		
10	null	null		
11	null	null		
12	null	null		
13	null	null		
14	null	null		Reserved for future [MRTD]
15	null	null		Similar to Active Authentication Public Key Info. Datagroup [MRTD], and the X.509 Certificate for PIV Authentication
16	null	null		Person(s) to notify

[MIT] Mandatory at time of instantiation.

Note: Data groups not implemented are grayed out.

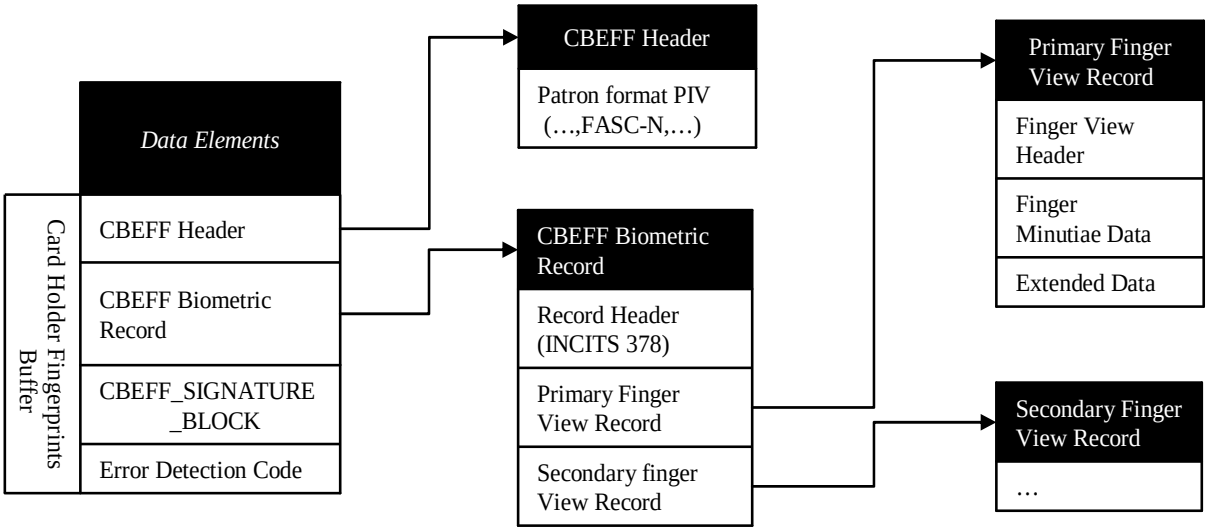
5.4 Card Holder Fingerprints Container (0x6010)

The Card Holder Fingerprints Container is mandatory and is implemented in accordance to SP800-73-1.

Notes regarding differences between sp800-73 and SP800-73v1 standards. SP800-73v1 finger minutiae changes:

- Primary and secondary fingerprints are stored in a single Card Holder Fingerprints container.
- Card Holder Fingerprints max size with minutia now 4K.

Figure 7. Card Holder Fingerprints Container Structure



Further details on this structure are specified on the SP800-76 table 3 and INCITS 378 Sec 6 table 6.

5.4.1 CBEFF Biometric Record Syntax

In accordance with SP 800-76, fingerprint minutiae and facial image are stored on the card. The data elements for these biometrics follow INCITS wrapped in a CBEFF. These are also stored in DMDC Personnel Data Repository (PDR). The two fingerprints (primary and secondary) are stored in a single container 0x6010.

PIV biometric data is embedded in a data structure conforming to Common Biometric Exchange Formats Framework [CBEFF]. This specifies that all biometric data shall be digitally signed and uniformly encapsulated. This covers: the PIV Card fingerprints mandated by [FIPS] and the Facial Image.

All such data is signed in the same manner as prescribed in [FIPS 201] and [800-73] for the biometric elements. The issuer signature is present for integrity and is stored in the CBEFF signature block. However, the certificate is in the CHUID. The overall arrangement of CBEFF and references is depicted in Table below.

Table 12. Simple CBEFF Structure

Data Element	References
CBEFF_HEADER	One instance of the CBEFF header (SP800-76 section 6 Table 7 Patron format PIV), and one instance of the "General Record Header" (INCITS 378 section 6.4), other references can be found in INCITS 398 5.2.1,
CBEFF_BIOMETRIC_RECORD	Two instances of the "Finger View Record" (INCITS 378 sec 6.5). The number of instances is indicated by the CBEFF Header "number of views".
CBEFF_SIGNATURE_BLOCK	One instance of the CBEFF Signature Block (SP800-76 Sec 6 Table 7), other references: FIPS 201 4.4.2, INCITS 398 5.2.3. CMS-compliant Issuer signature including FASC-N.

The SP800-76 template specification restricts the options of INCITS 378;

- No extended data.
- No proprietary data.
- Restriction of minutia type (bifurcation, ridge ending).

5.4.2 CBEFF Biometric Record

All fields for the biometric record are defined in INCTS 378 Sec 6. Additional explanations and sample data on the Minutiae record can be found in Appendix "*Sample data for "Card Holder Fingerprints" Container*" of this document. SP 800-76 specifies the required Patron Format CBEFF header which includes the FASC-N (see 800-76 p. 22).

5.4.3 CBEFF Signature Block

Details on the process for generating and populating the CBEFF Signature Block are described in FIPS201 sec 4.4.2 and SP800-73. The CBEFF Signature Block contains the CMS compliant issuer signature. The signature includes the FASC-N as a signed attribute.

As FIPS201 states in sec 4.4.2: "*If the signature on the biometric was generated with the same key as the signature on the CHUID, the certificates fields shall be omitted*".

5.5 Card Holder Facial Image Buffer (0x6030)

The Facial image is optional but DoD mandatory. It is not used for image processing but for biometric visual identification. As with the fingerprints it is wrapped in the CBEFF wrapper, and includes a CBEFF signature block with the FASC-N as a signed attribute. and is protected by the security object.

5.6 X.509 Certificates for PIV Authentication and CAC PKI Signature

The PIV Auth certificate is the only PIV mandatory certificate. The DoD Email Signing certificate is used in the Transitional PIV implementation for PKI logon, whereas the PIV Auth certificate is used in the PIV End-Point implementation. The DoD email certificate does not contain the FASC-N or the NACI.

The table below exhibits the access control rules for PIV and CAC certificate usage. "*NIST SP-800-73-1 Certificates"* lists the keys in use by the DoD today. "*CAC Certificates"* lists the optional and mandatory certificate (keys) proposed by the PIV standard. The PIV Authorization certificate is the only mandatory certificate. In the Transitional PIV

implementation for the DOD, the PIV Authentication certificate equates to the DoD email signing key pair and certificate.

Table 13. PIV, CAC Key, and Certificate Access Rules.

NIST SP-800-73-1 Certificates				
Key Name	***Key Purpose***	***Access Read Cert / Sign***	***OID***	***M / O***
PIV Authentication Certificate	Used to Authenticate the card and the CH using PIN. Identity key for logical access.	PIN/PIN	0x0101*	**M**
Digital Signature Certificate	Digital Signature for non-repudiation. Contact only	PIN/PIN-Always	0x0100	**O**
Key Management Certificate	Encryption key. Contact only	PIN/PIN not needed	0x0102	**O**
CAC Certificates				
PKI Signature Key	PKI Logical Login (Outlook) Digital Signature with non-repudiation, logical access, PIN. Outlook requires special extension.	ALW/PIN	0x0101	**M**
PKI Identity Key	Can be used for non repudiation signing outside Outlook.	ALW/PIN	0x0100	**M**
PKI Encryption Key	Key Encipherment	ALW/PIN	0x0102	**M**
Note: The gray area in the table indicates keys which the DoD functionally implemented using existing DoD PKI keys. * When the PIV Auth certificate is accessed via the Transitional card edge (PIV Auth activated mode) the OID is 0xA001.				

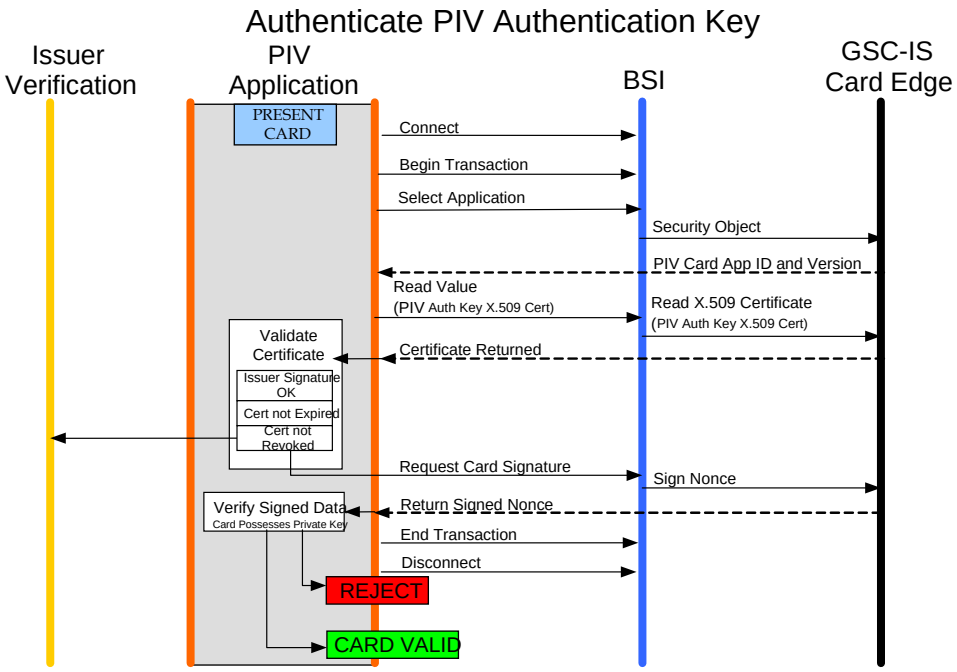
The PIV Auth certificate and the CAC PKI Signature certificate have similar functionality. The middleware must specifically differentiate between these keys.

5.7 PIV Authentication Certificate (0x0101)

This certificate is used to authenticate the card holder for logical access scenarios. The CAC PKI Signature key and associated certificate is used for Microsoft cryptographic logon and PKI signature. The DoD certificate does not include the NACI (as specified by FIPS 201), but it does contain the FASC-N. It also contains a UN = email address.

The diagram below illustrates the process of authenticating the PIV Authentication Certificate.

Figure 8. Authenticate PIV Authentication Key Sequence diagram



5.7.1 DoD PIV Auth Mode (Activated vs. Non-activated)

The DoD CAC End-Point supports both the Transitional PIV (for use by the DoD) and the End-Point PIV (for use outside the DoD). Although this should be transparent within and outside the DoD, it has affected the architecture of the card, introducing new terms. This section describes several details about the support of both PIV models on the CAC.

DoD implements the federally required PIV authentication certificate (with FASC-N) by adding a 4th PKI certificate to its existing three certificates. The DoD PIV authentication certificate has similar key usage as existing DoD PKI E-mail Signing Certificate (e.g. cryptographic Logon and client authentication). If both certificates are available to workstations, they may look identical and user confusion could result in the GSC-IS mode.

In order to avoid this confusion; DoD CAC PIV End-Point will have two modes that are configurable in GSC-IS:

- **DoD Non-activated PIV Auth Mode:** (Default CAC PIVs will be issued as such from RAPIDS).
  - Access 3 DoD certificates via PIV Transitional (GSC-IS) card edge
- **DoD Activated PIV Auth Mode:** (must be configured post issuance to this mode)
  - Access 3 DoD certificates plus 1 PIV Auth certificate via the PIV Transitional (GSC-IS) card edge.

The following table presents certificate access for Activated mode and Non-activated mode.

Table 14. PIV Auth Activated and Non-activated Mode

PIV Auth Non-activated mode	PIV Auth Activated mode
PKI ID Certificate	PKI ID Certificate
PKI Signature Certificate	PKI Signature Certificate
PKI Encryption Certificate	PKI Encryption Certificate
N/A*	PIV authentication Certificate

*N/A – Not accessible.

Relevant Use-Cases include the following:

- Most DoD users will be operating in default **DoD Non Activated PIV Auth Mode.**
- Few DoD users (e.g. AF personnel from DoD workstation) will use **DoD Activated PIV Auth Mode** to utilize CAC PIV End-Point to do the following on DoD workstations with GSC-IS Middleware:
  - Cryptographic logon remotely (e.g. VPN) to Non DoD (e.g. other Federal) networks using the Federal PIV Authentication Certificate
  - Client authentication to Non DoD (e.g. other Federal) web servers/portals using the Federal PIV Authentication Certificate

DoD CAC PIVs can be configured to **DoD Activated PIV Auth Mode** only through UMP-PIP[13].

Interaction with UMP-PIP to configure the default DoD CAC PIV to **DoD Activated PIV Auth Mode** will change the following:

- PIV Auth Certificate Object ID is added into Card Capability Container (CCC) for middleware visibility
- Update the Access Control Rule (ACR) of *Private Sign/Decrypt* Card Edge Command for PIV Auth Certificate in the Access Control Applet (ACA)

Note: To access the PIV Auth Certificate from the End-Point card edge, implementers first select the PIV EP applet and then issue a GET_DATA request as per SP800-73. To access the PIV Auth Certificate from the PIV Transitional card edge (PIV Auth Activated mode), implementers use OID 0xA001.

6 CONFORMANCE TESTING

6.1 CAC End-Point implementation conformance Testing

The DoD recognizes that NIST SP 800-85b is the vehicle for conformance testing of the PIV End-Point. DoD discrepancies are the, optional data elements, , non-compliant cameras that have not been tested against the FACESTD standard. The PIV Auth. Certificate test fails on not finding the certificate policy and the PIV interim extension.

[13] Implementers should be aware that middleware configuration settings may also be used to change these modes.

Appendix A Acronyms

The following terms are used throughout this document:

Authentication: Ensures that the individual is who he or she claims to be. This term is more about providing the evidence for this claim of authenticity.

Validation: The act of finding or testing the truth of something

Verification: Review process for determining or confirming the accuracy of information provided

proof that something that was believed (some fact or hypothesis or theory) is correct

Authorization: Access granted as a result of authentication and verification

The following abbreviations are used throughout this document:

ACO	Access Card Office
AID	Application Identifier
API	Application Programming Interface
BER	Basic Encoding Rules
BSI	Basic Services Interface
CAC	Common Access Card
CBEFF	Common Biometric Exchange File Format
CCC	Card Capabilities Container
CHUID	Card Holder Unique Identifier
DER	Distinguished Encoding Rules
DMDC	Defense Manpower Data Center
DoD	Department of Defense
DOSF	DMDC Open Specifications Framework
EF	Element Field
FASC-N	Federal Agency Smart Credential Number
GC	Generic Container
GSC-IS	Government Smart Card Interoperability Specification
GUID	Global Unique Identifier
NAC	National Agency Check
PIV	Personal Identity Verification
PIX	Proprietary Identifier Extension
PKI	Public Key Infrastructure

Appendix B References

[378] ANSI INCITS 378-2004, *Finger Minutiae Format for Data Interchange*, February 20, 2004

[CBEFF] Common Biometric Exchange File Format

[FIPS 201] NIST *Federal Information Processing Standards Publication 201-1, Personal Identity Verification for Federal Employees and Contractors*, March 2006. Updated June 26 2006.

[GP] *Open Platform, Card Specification*, v2.0.1', GlobalPlatform, April 2000

[GSC-IS] *Government Smart Card Interoperability Specification*, Version 2.1, NIST Interagency Report 6887 – 2003 Edition, July 16, 2003.

[HSPD-12] HSPD 12, *Policy for a Common Identification Standard for Federal Employees and Contractors*, August 27, 2004.

[JC] *Java Card 2.1.1 Platform Documentation*, Available from: http://java.sun.com/products/javacard/specs.html#211

[MRTD] *PKI for Machine Readable Travel Documents Offering ICC Read-Only Access,* Version - 1.1 Date - October 01, 2004. Published by authority of the Secretary General, International Civil Aviation Organization.

[NG] *DoD Implementation Guide for CAC Next Generation (NG)*, Version 2.6. Published by DoD CTIS Division.

[PACS 2.2] *Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems*, Version 2.2, The Government Smart Card Interagency Advisory Board's Physical Security Interagency Interoperability Working Group, July 27, 2004.

[PACS 2.3] *Draft Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems*, Version 2.3, The Government Smart Card Interagency Advisory Board's Physical Security Interagency Interoperability Working Group, August 9, 2005.

[PCSC] Personal Computer/Smart Card Workgroup Specifications*, Interoperability Specification for ICCs and Personal Computer Systems*, Revision 2.01, 2005.

[SP800-73-1] NIST Special Publication 800-73-1, *Integrated Circuit Card for Personal Identity Verification*, NIST, March 2006.

[SP800-73 Errata] Errata for NIST Special Publication 800-73, *Errata for SP800-73-1*, May 2 2006.

[SP800-76] NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification,* NIST, February 1, 2006.

[SP800-76 Errata] Errata for NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification,* updated, July 19, 2006.

[SP800-78] NIST Special Publication 800-78, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification,* NIST, March 2005.

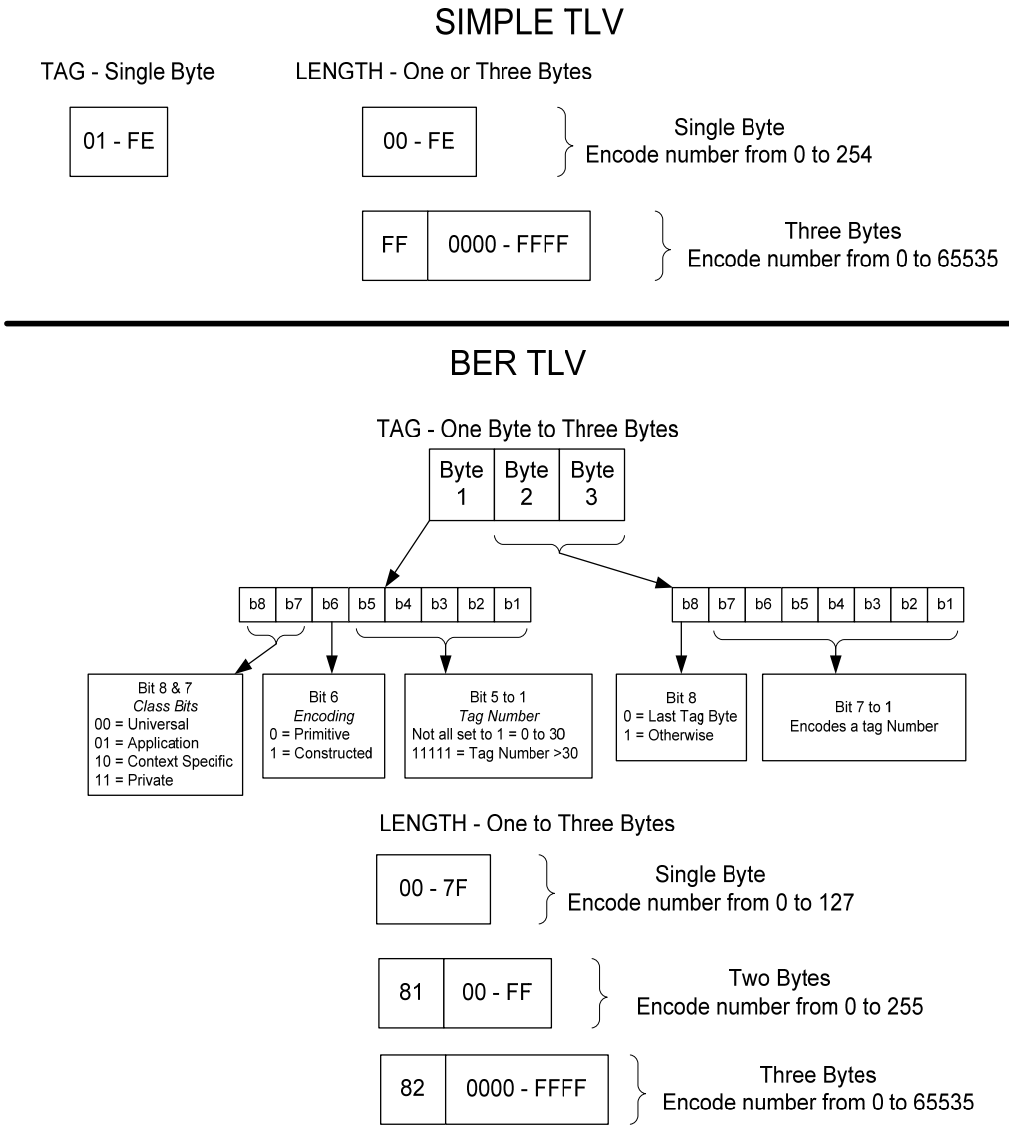
[SP800-79] NIST Special Publication 800-79, *Guidelines for the Certification and Accreditation of PIV Card Issuing Organizations*, NIST, July 2005.

[SP800-85B] NIST Special Publication 800-85, *Draft NIST Special Publication 800-85, PIV Data Model Test Guidelines,* NIST, July 2006.

[SP800-87] NIST Special Publication 800-87, Draft NIST Special Publication 800-87, Codes for the Identification of Federal and Federally-Assisted Organizations, NIST, August 2005.

Appendix C Comparison of Simple TLV and BERT TLV.

PIV Transitional is encoded with simple TLV. PIV end-point specifies the use of Basic Encoding Rules (BER-TLV). Both are defined in ISO 7816-4. For informational purposes, the difference between **generic** SIMPLE and BER is shown in the figure below. PIV End-Point implements Table 5, which is non-compliant in that it has 1 byte tags for backward compatibility.



Appendix D Sample data for PIV Authentication Certificate

Below is a sample record format for the DoD PIV Authentication Certificate.

Summary: Sample PIV Auth Certificate (2048 Root, 2048 Intermediate, and 1024 End Entity) issued from JITC CA 19 on Jan of 2008. The sequence TBSCertificate contains information associated with the subject of the certificate and the CA who issued it. Every TBSCertificate contains the names of the subject and issuer, a public key associated with the subject, a validity period, a version number, and a serial number; some MAY contain optional unique identifier fields. For signature calculation, the data that is to be signed is encoded using the ASN.1 distinguished encoding rules (DER) [X.690]. ASN.1 DER encoding is a tag, length, value encoding system for each element.

Sources: *ASN.1, **ASN.1. Specified in 800-73 for interoperable use**ASN.1. Specified in 800-73 for interoperable use

Field Name	Hex Raw Data	OID	Data Content
Basic Certificate			
*Version	A0 03 02 01 02		2
*Serial Number	02 02 13 03		4867
** Issuer Signature Algorithm	30 0D 06 09 **2A 86 48 86 F7 0D 01 01 05** 05 00	2A 86 48 86 F7 0D 01 01 05 (1.2.840.113549.1.1.5 =SHA1withRSA)	SHA1withRSA Encryption
* Issuer Distinguished Name	30 5C 31 0B 30 09 06 03 **55 04 06** 13 02 55 53 31 18 30 16 06 03 **55 04 0A** 13 0F 55 2E 53 2E 20 47 6F 76 65 72 6E 6D 65 6E 74 31 0C 30 0A 06 03 **55 04 0B** 13 03 44 6F 44 31 0C 30 0A 06 03 **55 04 0B** 13 03 50 4B 49 31 17 30 15 06 03 **55 04 03** 13 0E 44 4F 44 20 4A 49 54 43 20 43 41 2D 31 39	55 04 06 (2.5.4.6 =**countryName**)	'US'
		55 04 0A (2.5.4.10=**organizationName**)	'U.S. Government'
		55 04 0B (2.5.4.11= **organizationalUnitName**)	'DoD'
		55 04 0B (2.5.4.11= **organizationalUnitName**)	'PKI'
		55 04 03 (2.5.4.3 =**commonName**)	'DOD JITC CA-19'
*Validity Period	30 1E 17 0D 30 38 30 31 31 34 30 30 30 30 30 5A		080114000000Z UTC Time Code(**issued date**)
	17 0D 31 31 30 31 31 30 32 33 35 39 35 39 5A		110110235959Z UTC Time Code (**expiration date**)
*Subject Distinguished Name	30 81 90 31 0B 30 09 06 03 **55 04 06** 13 02 55 53 31 18 30 16 06 03 **55 04 0A** 13 0F 55 2E 53 2E 20 47 6F 76 65 72 6E 6D 65 6E 74 31 0C 30 0A 06 03 **55 04 0B** 13 03 44 6F 44 31 0C 30 0A 06 03 **55 04 0B** 13 03 50 4B 49 31 0D 30 0B 06 03 **55 04 0B** 13 04 55 53 41 46 31 3C 30 3A 06 03 **55 04 03** 13 33 4F 43 53 20 50 49 56 41 55 54 48 20 59 45 53 2E 4D 49 4C 4F 4E 2E 43 41 52 44 20 54 48 52 45 45 20 50 49 56 42 45 54 41 2E 31 34 30 31 33 35 37 33 35 38	55 04 06 (2.5.4.6=**countryName**)	'US'
		55 04 0A (2.5.4.10 =**organizationName**)	'U.S. Government'
		55 04 0B (2.5.4.11=**organizationalUnitName**)	'DOD'
		55 04 0B (2.5.4.11=**organizationUnitName**)	'PKI'
		55 04 0B (2.5.4.11=**organizationUnitName**)	'USAF'
		55 04 03 (2.5.4.3 =**commonName**)	'OCS PIVAUTH YES.MILON.CARD THREE PIVBETA.1401357358'
*Subject Public Key Information	30 81 9F 30 0D 06 09 **2A 86 48 86 F7 0D 01 01 01**	2A 86 48 86 F7 0D 01 01 01 (1.2.840.113549.1.1.1= RSA)	RSA Encryption
**Public Key	03 81 8D 00 30 81 89 02 81 81 00 B7 BE 11 EC B0 3D B3 76 4A 04 1F B8 2A 00 2E C1 C5 3F EF 40 73 5D 4D ED E1 E4 CE E4 49 8F 8D 98 16 3F BE D1 8C 97 05 9C BA 96 E1 8C E8 00 EA 6A A8 95 12 FC 7A 67 83 34 82 BF A7 6B 4B 20 AC 49 BA 73 25 8C 8D 90 A5 BF ED 4E DD 17 49 50 E6 BB E2 5E 0A A2 19 C4 4E 22 4F A3 8D 50 56 25 0F 33 97 E9 C9 CF 6E 33 8D 32 DF EE 16 38 F1 FD 1A 62 9B 90 A6 39 54 1A DC F8 F8 D6 50 B7 58 20 1A F1 02 03 01 00 01		

Standard Extensions			
*Authority Key Identifier	A3 82 01 C8 30 82 01 C4 30 1F 06 03 **55 1D 23** 04 18 30 16 80 14 A9 56 F9 48 50 1D FF 45 1C 54 E2 63 6B FA 71 2C 98 34 C8 8C	55 1D 23 (2.5.29.35=**authorityKeyIdentifier**)	
		80 14 A9 56 F9 48 50 1D FF 45 1C 54 E2 63 6B FA 71 2C 98 34 C8 8C	Key Value
*Subject Key Identifier	30 1D 06 03 **55 1D 0E** 04 16 04 14 27 FF A1 A1 1D 41 3B F0 1C DF 71 9D BC 57 DF 1C 24 47 B4 87	55 1D 0E (2.5.29.14 =**subjectKeyIdentifier**)	
		27 FF A1 A1 1D 41 3B F0 1C DF 71 9D BC 57 DF 1C 24 47 B4 87	Key Value
*Extended Key Usage	30 1F 06 03 **55 1D 25** 04 18 30 16 06 0A **2B 06 01 04 01 82 37 14 02 02** 06 08 **2B 06 01 05 05 07 03 02**	55 1D 25 (2.5.29.37 =**extKeyUsage**)	
		2B 06 01 04 01 82 37 14 02 02 (1.3.6.1.4.1.311.20.2.2**= kp_smartCardLogin**)	
		2B 06 01 05 05 07 03 02 1.3.6.1.5.5.7.3.2= **id_kp_clientAuth**)	
*Certificate Policies	30 16 06 03 **55 1D 20** 04 0F 30 0D 30 0B 06 09 **60 86 48 01 65 02 01 0B 09**	55 1D 20 (2.5.29.32 =**certificatePolicies**)	
		60 86 48 01 65 02 01 0B 09 (2.16.840.1.101.2.1.11.9 =null)	
*Subject Alternative Name	30 58 06 03 **55 1D 11** 04 51 30 4F A0 24 06 0A **2B 06 01 04 01 82 37 14 02 03** A0 16 0C 14 **31 34 30 31 33 35 37 33 35 38 31 35 37 30 30 34 40 6D 69 6C** A0 27 06 08 **60 86 48 01 65 03 06 06** A0 1B 04 19 D4 F8 10 DA 01 15 6C 10 C0 88 85 83 60 DA 04 0C 33 5E 66 A2 85 78 10 93 F0	55 1D 11 (2.5.29.17=**subjectAltName**)	
		2B 06 01 04 01 82 37 14 02 03 (1.3.6.1.4.1.311.20.2.3 =**nt_userPrincipalName**)	
		31 34 30 31 33 35 37 33 35 38 31 35 37 30 30 34 40 6D 69 6C	'1401357358157004@mil'
		60 86 48 01 65 03 06 06 (2.16.840.1.101.3.6.6 =**pivFASC-N**)	D4 F8 10 DA 01 15 6C 10 C0 88 85 83 60 DA 04 0C 33 5E 66 A2 85 78 10 93 F0
*Subject Directory Attributes	30 1B 06 03 **55 1D 09** 04 14 30 12 30 10 06 08 **2B 06 01 05 05 07 09 04** 31 04 13 02 55 53	55 1D 09 (2.5.29.9 =**subjectDirectoryAttributes**)	
		2B 06 01 05 05 07 09 04 1.3.6.1.5.5.7.9.4 (**id-pda-countryOfCitizenship**)	'US'
*Authority Info Access	30 7E 06 08 **2B 06 01 05 05 07 01 01** 04 72 30 70 30 3E 06 08 **2B 06 01 05 05 07 30 02** 86 32 68 74 74 70 3A 2F 2F 63 72 6C 2E 6E 69 74 2E 64 69 73 61 2E 6D 69 6C 2F 67 65 74 73 69 67 6E 3F 44 4F 44 25 32 30 4A 49 54 43 25 32 30 43 41 2D 31 39 30 2E 06 08 **2B 06 01 05 05 07 30 01** 86 22 68 74 74 70 3A 2F 2F 6F 63 73 70 2E 6E 73 6E 30 2E 72 63 76 73 2E 6E 69 74 2E 64 69 73 61 2E 6D 69 6C	2B 06 01 05 05 07 01 01 (1.3.6.1.5.5.7.1.1=**id_pe_authorityInfo Access**)	
		2B 06 01 05 05 07 30 02 (1.3.6.1.5.5.7.48.2=(**id_ad_caIssuers**)	http://crl.nit.disa.mil/getsign?DOD%20JITC%20CA-19
		2B 06 01 05 05 07 30 01 (1.3.6.1.5.5.7.48.1=**ocsp**)	http://ocsp.nsn0.rcvs.nit.disa.mil
*CRL Distribution Points	30 42 06 03 **55 1D 1F** 04 3B 30 39 30 37 A0 35 A0 33 86 31 68 74 74 70 3A 2F 2F 63 72 6C 2E 6E 69 74 2E 64 69 73 61 2E 6D 69 6C 2F 67 65 74 63 72 6C 3F 44 4F 44 25 32 30 4A 49 54 43 25 32 30 43 41 2D 31 39	55 1D 1F (2.5.29.31=**cRLDistributionPoints**)	
		68 74 74 70 3A 2F 2F 63 72 6C 2E 6E 69 74 2E 64 69 73 61 2E 6D 69 6C 2F 67 65 74 63 72 6C 3F 44 4F 44 25 32 30 4A 49 54 43 25 32 30 43 41 2D 31 39	http://crl.nit.disa.mil/getcrl?DOD%20JITC%20CA-19
*Key Usage	30 0E 06 03 **55 1D 0F** 01 01FF 04 04 03 02 07 80	55 1D 0F (2.5.29.15=(**keyUsage**)	
		01 01	Boolean type, length
		FF	TRUE -Critical
		03 02 07 80	Digital Signature

End of Standard Extensions			
**Digital Signature	30 0D 06 09 **2A 86 48 86 F7 0D 01 01 05** 05 00 03 82 01 01 00 74 C1 C6 93 97 E4 5B 61 10 40 BE FA A5 01 0F 2B 08 26 A2 89 B5 82 B9 FF 53 FB 24 01 78 02 95 00 E9 90 2D 62 3F 0C 7D AD 92 0F D8 AC 48 BB E6 D7 DF DA BA 9B BA C9 F5 67 75 63 C6 A1 63 4D 46 E8 EA 58 54 3B 48 7F 2C 80 78 4A D1 6F E3 AA A2 84 29 88 9F 1F FA 5E 4C 05 90 F3 32 E0 DB EB C8 D0 A8 7E 08 DE 29 EE F2 0A 02 81 BB 16 9C 90 AF BD 2A F5 D3 0C 6E CF 1A D8 AB 7E 7D 67 42 3F F0 62 33 C2 14 99 DF 9C E1 CA 28 0B 9E CA 4B FF 97 67 9B 57 F6 21 7F 40 58 14 F9 A0 D3 7C B7 CC 52 A7 2C 96 CD 8A D3 3B 05 0A 51 73 F9 0D 6C 94 0B 77 88 1C 6F 4C F6 69 A0 D8 5D D7 C8 80 73 43 A8 E0 46 F7 32 32 CF A8 47 DD D1 E1 9F C3 AC 46 FB 43 06 74 91 59 75 16 70 29 A6 30 A9 73 CF C1 D5 D7 C3 F7 75 24 86 7E 87 BE D3 0F 54 03 AB 10 C5 48 1B BE 3B 34 56 F1 D9 4D 50 33 6C C6 F0 97 23 58	2A 86 48 86 F7 0D 01 01 05 1.2.840.113549.1.1.5 (SHA1withRSA)	SHA1withRSA Encryption followed by signature.

Example of Data Encoding: Reference ITU X.690 02-07

Issuer Algorithm	Identifier Octet (section 8.1.2)	Length Octet (section 8.1.3)	Contents Octet (section 8.1.4)	# of OID Bytes	1.2.840.113549.1.1.5 (SHA1withRSA) Dotted Decimal Representation of OID
Raw Data	30	0D	06	09	2A 86 48 86 F7 0D 01 01 05 (OID)

Version 1.10

Appendix E DoD CAC PIV Transitional and End-Point Quick Guide

DoD CAC PIV Transitional and End-Point SP 800-73-v1

Buffer Description	ContainerID	Maximum Length (Bytes)	Access Rule	Contact /Contactless	M/O
Card Capabilities Container	0xDB00	266	Always Read	Contact	M
Card Holder Unique Identifier	0x3000	3377	Always Read	Contact and Contactless	M
X.509 Certificate for PIV Authentication	0xA001	1651	PIN	Contact	M
Card Holder Fingerprints	0x6010	7768	PIN	Contact	M
Card Holder Facial Image	0x6030	12704	PIN	Contact	O
X.509 Certificate for Digital Signature	0x0100	1651	PIN Always	Contact	O
X.509 Certificate for Key Management	0x0102	1651	PIN	Contact	O
X.509 Certificate for Card Authentication	0x0500	1651	Always	Contact and Contactless	O
Security Object	0x9000	1000	Always Read	Contact	M

Card Capabilities Container *		0xDB00	Always Read
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Card Identifier	0xF0	Fixed	21
Capability Container version number	0xF1	Fixed	1
Capability Grammar version number	0xF2	Fixed	1
Applications CardURL	0xF3	Variable	128
PKCS#15	0xF4	Fixed	1
Registered Data Model number	0xF5	Fixed	1
Access Control Rule Table	0xF6	Fixed	17
CARD APDUs	0xF7	Fixed	0
Redirection Tag	0xFA	Fixed	0
CapabilityTuples (CTs)	0xFB	Fixed	0
StatusTuples(STs)	0xFC	Fixed	0

Card Holder Unique Identifier *		0x3000	Always Read
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
FASC-N	0x30	Fixed Text	25
GUID	0x34	Fixed Numeric	16
Expiration Date	0x35	Date (YYYYMMDD)	8
Authentication Key Map (Optional)	0x3D	Variable	512
Issuer Asymmetric Signature	0x3E	Variable	2816
Error Detection Code	0xFE	LRC	0

X.509 Certificate for PIV Authentication		0x0101	pkiCompute PIN
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Certificate	0x70	Variable	1856
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Card Holder Fingerprints		0x6010	PIN
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Fingerprint I and II	0xBC	Variable	2000
Error Detection Code	0xFE	LRC	0

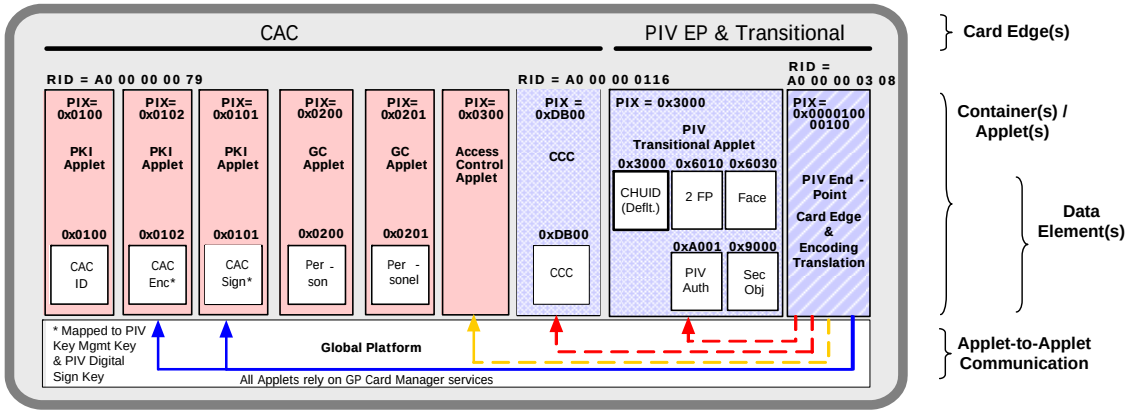
Security Object		0x9000	Always Read
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Mapping of DG to ContainerID	0xBA	Variable	100
Security Object (Issuer Signature)	0xBB	Variable	900
Error Detection Code	0xFE	LRC	0

NIST SP-800-73					CAC				
Key Name	Key Purpose	Access Read Cert / Sign	OID	M / O	Key Name	Key Purpose	Access Read Cert / Sign	OID	M / O
PIV Authentication Key	Used to Authenticate the card and the cardholder using PIN. Identity key for logical access.	PIN/PIN	0x0101	M	PKI Signature Key	PKI Logical Login (Outlook) Digital Sign with non-repudiation, logical access, PIN. Outlook requires special extension.	ALW/PIN	0x0101	M
Digital Signature Key	Digital Sign for nonrepudiation Contact only	PIN/PIN-Always	0x0100	O	PKI Identity Key	Can be used for non repudiation signing outside Outlook.	ALW/PIN	0x0100	M
Key Management	Encryption key. Contact only	PIN/PIN not needed	0x0102	O	PKI Encryption Key	Key Encipherment	ALW/PIN	0x0102	M

FASC-N (10 BCD Digits)			
Field Name	L (BCD)	Field description	
AGENCY CODE	4	Identifies the government agency issuing the credential	Concaten. Unique Per card
SYSTEM CODE	4	Identifies the system the card is enrolled in, is unique for each site	
CREDENTIAL NUMBER	6	Encoded by the issuing agency. For a given system no duplicate numbers are active	
CS	1	CREDENTIAL SERIES (SERIES CODE) Field is available to reflect major system changes	
ICI	1	INDIVIDUAL CREDENTIAL ISSUE (CREDENTIAL CODE) Initially encoded as "1", will be incremented if a card is replaced due to loss or damage	
PI	10	PERSON IDENTIFIER Numeric Code used by the identity source to uniquely identify the token carrier. (e.g. DoD EDIPI)	DoD EDIPI
OC	1	ORGANIZATIONAL CATEGORY 1- Federal Government Agency 2- State Government Agency 3- Commercial Enterprise 4- Foreign Government	Static for given card holder
OI	4	ORGANIZATIONAL INDENTIFIER OC=1 – FIPS 95-2 Agency Code OC=2 – State Code OC=3 – Company Code OC=4 – Numeric Country Code	
POA	1	PERSON/ORGANIZATION ASSOCIATION CATEGORY 1 – Employee 2 – Civil 3 – Executive Staf 4 – Uniformed Service 5 – Contractor 6 – Organizational Affiliate 7 – Organizational Beneficiary	
SS	1	Start Sentinel. Leading character which is read first when card is swiped	
FS	1	Field Separator	
ES	1	End Sentinel	
LRC	1	Longitudinal Redundancy Character	

Get Data
Verify/Change Pin
Sign

DoD Default Container



1. CHUID signature BER/TLV
2. Transitional PIV CHUID encoded BER/TLV except tags 2 bytes for BC
3. PIV Auth cert OID is 0x101 if accessed through End-Point card edge, and 0xA001 if accessed through the PIV Transitional in PIV Auth Activated mode.

DMDC CTIS/Technical Advisory Group (TAG) 6-14-08

Appendix F Sample Java program: Accessing the CHUID

Below, is a sample Java program accessing the CHUID using NIST SP800-73-1 Card edge APDU commands. The program was developed using Java 6.0 and it's Java Card Smart Card I/O package ("javax.smartcardio.*;"). The local host program assumes a standard PIV End-Point smart card.

```
// BEGINNING OF SAMPLE PROGRAM ****************************************
/*
 * smartcardio test program, on PIV CHUID
 * @author Jonathan Arana, CAC Test Lab
 *
 * DISCLAIMER: This software is released by the CTL as a service and is
 * expressly provided "AS IS." This software was developed for demonst-
 * rative or educational purposes. The author and CTL MAKE NO WARRANTY
 * OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY, INDCLUDING, WITHOUT LIM-
 * ITATION, THE IMPLIED WARANTY OF MERCHANTABILITY, FITNESS FOR A PART-
 * ICULAR PURPOSE, NON INFRINGEMENT AND DATA ACCURACY. THE AUTHOR AND
 * CTL DO NOT REPRESENT OR WARRANT THAT THE OPERATION OF THIS SOFTWARE
 * WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ANY DEFECTS WILL BE
 * CORRECTED.
*/

import java.io.*;
import java.util.*;
import javax.smartcardio.*;
import java.lang.String;

public class Test3{

   /*
    * Create an instance of Test3 Class.
    */
   public Test3(){
   }

   /*
    * Main program driver.
    *       No arguments needed.
    */
   public static void main(String[] args)
   {
      System.out.println("          ------------------------------" );
      System.out.println("          ====Java Smartcard IO Test====");

      try{
            smartIoApiEx();     // Setup communication between SC and local
Host PC.

      }catch(CardException e){
            System.out.println("Error occured in transaction: "+e);
      }
      return ;
   }

   /*
```

```
    * Java (smartcardio) smart card connection/communication setup
    */
   public static void smartIoApiEx() throws CardException{
      //the AID for PIV Application
      byte[] aid = {(byte)0xA0, 0x00, 0x00, 0x03, 0x08, 0x00, 0x00, 0x10,
0x00};
      // Response buffer.
      String response;

      //show the list of available terminals
      TerminalFactory factory = TerminalFactory.getDefault();
      List<CardTerminal> terminals = factory.terminals().list();
      System.out.println("          ------------------------------" );
      System.out.println("Terminals: " + terminals);

      //get the first terminal
      System.out.println("Obtaining terminal 0");
      CardTerminal terminal = terminals.get(0);

      //establish a connection with the card
      System.out.println("          ------------------------------" );
      System.out.println("Connecting to card");
      Card card = terminal.connect("T=0");
      System.out.println("card: " + card);
      CardChannel channel = card.getBasicChannel();

      //execute command APDUs to attain CHUID
      getChuid(aid, channel);

      //disconnect
      card.disconnect(false);
      return;
   }

   /*
    * Obtain the CHUID using the PIV Interface
    */
   private static void getChuid(byte[] cert, CardChannel channel)
         throws CardException{
      ResponseAPDU res;
      int nr = 0;

      // select the app
      select(channel, cert);
      // get the data
      res = getData(channel);

      //output data (e.g. console, file, etc.)
      System.out.println("          ------------------------------" );
      System.out.println("CHUID: ");
      nr = res.getNr();
      try{
         System.out.print(toString(res.getBytes(), nr));
      }catch(UnsupportedEncodingException e){
         System.out.println("Unsupported Encoding Error");
      }
```

```
}

/*
 * obtain the data of the currently selected app
 */
private static ResponseAPDU getData(CardChannel channel)
      throws CardException{
   // SP800-73-1 APDU GET_DATA APDU command.
   byte[] apdu={0x00, (byte)0xCB, 0x3F,
         (byte)0xFF, 0x05, 0x5C, 0x03,
         (byte)0x5F, byte)0xC1, 0x02}; // BERT-TLV CHUID tag
   int size, sw1, sw2, nr, count=0, offset;
   ResponseAPDU res;

   //obtain the first 2 bytes to get buf size
   res = channel.transmit(new CommandAPDU(apdu));
   //display status
   System.out.println("          ------------------------------" );
   System.out.println("GetData " + res.toString());

   return res;

}

/*
 * Select a card application/container
 */
private static void select(CardChannel channel, byte[] data)
      throws CardException{
   byte cla = 0x00,
        ins = (byte)(0xA4 & 0xFF),
         p1 = 0x04,
         p2 = 0x00,
         de = 0x00;
   int  sw1 = 0, sw2 = 0, nr = 0;

   ResponseAPDU res = channel.transmit(
               new CommandAPDU(cla, ins, p1, p2, data, de)
               );

   nr = res.getNr();
   //display status
   System.out.println("          ------------------------------" );
   System.out.println("Select " +  res.toString());
   try{
      System.out.println(toString(res.getBytes(), nr+2));
    //+2 for status words (90 00)
   }catch(UnsupportedEncodingException e){
      System.out.println("Unsupported Encoding Error");
   }

}

/*
 * convert a byte array to ASCII Hexadecimal
 */
private static String toString(byte[] buf, int length)
```

```
         throws UnsupportedEncodingException{
      String str;
      char[] chr = new char[(length * 2)];


      for(int i=0; i < length; i++){
         int h = (buf[i] & 0xF0) >> 4;
            int l = buf[i] & 0x0F;
            char c=0;

            if(h < 0x0A){
                  chr[i*2] = (char)(h + '0');
            }
            else{
                  chr[i*2] = (char)(h - 10 + 'A');
            }
            if(l < 0x0A){
                  chr[(i*2)+1] = (char)(l + '0');
            }
            else{
                  chr[(i*2)+1] = (char)(l - 10 + 'A');
            }
      }
      return new String(chr);
   }

} // END OF SAMPLE PROGRAM ****************************************
```

Output from program execution:


Command Prompt
D:\javaProjects\smartcard\PIVInterface_proof-of-concept>javac Test3.java
D:\javaProjects\smartcard\PIVInterface_proof-of-concept>java Test3

====Java Smartcard IO Test====

Terminals: [PC/SC terminal ActivCard USB Reader V2 0]
Obtaining terminal 0

Connecting to card
card: PC/SC card in ActivCard USB Reader V2 0, protocol T=0, state OK

Select ResponseAPDU: 26 bytes, SW=9000
61164F0BA0000003080000100001079074F05A0000003089000

GetData ResponseAPDU: 1616 bytes, SW=9000

CHUID:
5382064A3019D4F810DA01156C10C2A2858360DA040C32D14F8184388E13FC3410303030303030303030
3030303030303030303035083230313130333034 3E82060D308206090609 2A864886F70D010702A082
05FA308205F6020103310930070605 2B0E03021A300A06086086480165030601A08204233082041F
30820388A003020102020219E1300D06092A864886F70D0101050500305C310B3009060355040613
025553311830160603550 40A130F552E532E20476F7665726E6D656E74310C300A060355040B1303
446F44310C300A060355040B1303504B49311730150603550403130E444F44204A4954432043412D
3133301E170D3037303132323139323834345A170D3132303132313139323834345A3062310B3009
060355040613025553311830160603550 40A130F552E532E20476F7665726E6D656E74310C300A06
0355040B1303446F44310C300A060355040B1303504B49311D301B0603550403131446F44204361
726420497373756572205430303730819F300D06092A864886F70D010101050003818D0030818902
818100D4C2B97DD4F37C1ABD573CA4286B2058DE9B2667DDE1924A01140673DB30C8ED285663E856
3475ECC3EE11C0B0A028DD6E88BFB8A9B8964CBB40E5F81E1EA5B342AA2298F557F44905C6709B37
326B47D66D8C313D1965BAE33F2BAFD34FE0571FDDB100143DEB8ED5D14B297C83EEF2CB8771E8E6
40342AC7F74AED982D69D50203010001A38201E8308201E4301F0603551D23041830168014DB88CB
96504EFACD4D1C7EA94E2B4FC173D37917301D0603551D0E041604147739 88D209E69D889D4E1C7F
28C99D5912976A7A30130603551D25040C300A06086086480165030607300E0603551D0F0101FF04
04030206C030160603551D20040F300D300B06096086480165020 10B093081DF0603551D1F0481D7
3081D4303BA039A037863568747470 3A2F2F63726C2E6764732E6E69742E646973612E6D696C2F67
657463726C3F444F442532304A4954432532304 3412D3133308194A08191A0818E86818B6C646170
3A2F2F63726C2E6764732E6E69742E646973612E6D696C2F636E253364444F442532304A49544325
323043412D31332532636F75253364504B492532636F75253364446F442532636F253364552E532E
253230476F7665726E6D656E74253263632533645553 3F6365727469666963617465726 5766F6361
74696F6E6C6973743B62696E617279308182060 82B0601050507010104763074304206082B060105
0507300286366874747 03A2F2F63726C2E6764732E6E69742E646973612E6D696C2F676574736967
6E3F444F442532304A4954432532304 3412D3133302E06082B0601050507300186226874747 03A2F
2F6F6373702E6E736E302E726376732E6E69742E646973612E6D696C300D06092A864886F70D0101
0505000381810 0B631481DF3DD55C2A782A919FB464A750633873702A30F3E7DFC1C85EADA1F1A4E
562B93A96CAFD8613EC3B9A15A290452937 10D304CDAB04DF054780F867A61910433A22259CE0E37
0A9445C4145C25BE5837D0A942418CE3B2215028AC03EC40FF26B372D5F340754DE6787CA3A535F5
0AFE854C2BC9CBF881426A17D7B5A0318201B1308201AD020101306230 5C310B3009060355040613
025553311830160603550 40A130F552E532E20476F7665726E6D656E74310C300A060355040B1303
446F44310C300A060355040B1303504B49311730150603550403130E444F44204A4954432043412D
3133020219E1300706052B0E03021AA081AA30110603672A00310A060860864801650306013023 06
092A864886F70D0109043116041 4EDD007A710158C668AF66028D73421FC52B46C043070060860 86
4801650306053164306231 0B3009060355040613025553311830160603550 40A130F552E532E2047
6F7665726E6D656E74310C300A060355040B1303446F44310C300A060355040B1303504B49311D30
1B0603550403131 4446F442043617264204973737565722054303037300B06092A864886F70D0101
0504818080162C296A7DF641EE7E141C2BC29B416095F96FB77A7A959E69C4438DFDC700E9128E8D
050DD6AFF8194AB3C4E61B7EB9354B8F3773DC142F7F57E9E1E2443CBFAA3E94ADA724B1BFCA4AF9
E1C2B1CB021B3BD1AECD9AF6366CD2BE95CF5C64808879D67942CF4B18624F4763F9BBD713D04CDD
09AFE48C62AF4235BCEB3084FE00
D:\javaProjects\smartcard\PIVInterface_proof-of-concept>

Appendix G Sample data for "Cardholder Fingerprints" Container

This is a sample record format for the "*Card Holder Fingerprints*" container. This sample contains two fingers; one left index and a right index finger data. See INCITS 378 table 6 for minutiae record descriptions, and SP800-73-1 Appendix A.

		Sample Data (in Hexadecimal)	Description
CBEFF HEADER	Patron PIV Format	03	Patron Header Version
		0D	SBH Security Options 0x0D=signed not encrypted
		00000242	BDB length 27c4=10180=10134+14+20+12 010A=266=26 + 1*(4 + 39*6 + 2)
		01E5	SB length
		001B	format owner
		0201	format type
		140607141517325A	creation date
		140607141517325A	valid date start
		141007141517325A	valid date end
		000008	bio type
		80	bio data type
		FD	bio quality
		555320444F442052415049 44530000000000	creator (US DOD RAPIDS)
		D22010DA010C2D00843C0 D8360DA010842108430822 01093EB	FASC-N
		00000000	reserved
	General Record Header	464D5200	Format Id 'FMR'
		20323000	Version ' 20'
		0242	record length 5E=94=26+2(4+5*6)
		000C0A50	CBEFF product id
		0000	Capture equipment compliance and id
		0168	width in pixels (ox124=292)
		0168	height in pixels
		00C5	horizontal resolution pixel/cm (C5=197)
		00C5	vertical resolution
		02	number views (number of fingers)
		00	reserved
BIOMETRIC RECORD(S) …	Finger View	07	Position (left index finger)
		00	view number and impression
		FE	finger quality
		33	Number of minutiae (55 minutiae data records follow).
	Minutiae	8103005F3600	minutiae data
		810A00762D00	minutiae data
		….	….minutiae data
		….	….minutiae data

		0000	extended data (x00000 = none)
	Finger View	02	Position (right index finger).
		00	view number and impression
		FE	finger quality
		27	Number of minutiae (39 minutiae records).
	Minutiae	408F00A98900	minutiae data
		…	minutiae data
		…	minutiae data
		810700DE2F00	minutiae data
		80FB01078500	minutiae data
		…	minutiae data
		0000	extended data (x00000 = none)
CSB		55555555555555555555555….	CBEFF SIGNATURE BLOCK (CSB)

Appendix H PIV Data Encoding

The data content of a BER-TLV data object may consist of other BER-TLV data objects.

The PIV End-Point Data objects are BER-TLV objects encoded as per ISO/IEC 8825-2, except that tag values (T-values) of the PIV data object's inner tags do not conform to generic BER-TLV requirements and are 1 byte. This is due to the need to accommodate legacy tags inherited from the GSC-IS specification.

Thus, When the CAC End-Point responds to a PIV call for the CHUID from either the contactless or the contact interface, the CAC will return the following:

```
|Tag1(Simple)|Length1(BER)|Value1 |Tag2(Simple)|Length2(BER)|Value2|...
```

All container data elements are stored in BER-TLV format in a unique buffer, as follows. Each BER-TLV data object consists of a tag field (1 byte), a length field (from 1 to 3 bytes) and an optional value field. The Value field associated to each tag is appended after the T-L field itself, i.e. the PIV Data-Model content is seen as a list of successive BER-TLV.

The following figure shows the PIV Data-Model representation and the way data is returned by the PIV applet on response to GET DATA APDU.

Figure: GET DATA APDU sample response

Tag1 (1 byte)	Len1 (1 byte)	Value1 (1 byte)	Tag2 (1 byte)	Len2 (3 bytes) (0x82,lenH2,lenL2)	Value2 (2 byte)	Tag3 (1 byte)	Len3 (2 bytes) (0x81, len3)	Value3 (3 bytes)

The Len bytes may be in the range of 1 to 3 bytes depending on the value buffer size.

The applet enforces a minimal encoding of length field when returning data to the middleware. As a result, it applies the following rules on length field:

Table: BER-TLV Length Fields Encoding

Number of bytes	1st byte	2nd byte	3rd byte	N (nb of bytes in the value field)
1	‘00’ to ‘7F’	-	-	0 to 127
2	‘81’	‘80’ to ‘FF’	-	128 to 255
3	‘82’	‘0100’ to ‘FFFF’		256 to 65535

Appendix I Addressing of Data Objects

The addressing schemes specified for CAC (NISTIR 6887) and PIV are the same. Some terms used frequently in discussions of object addressing are defined below.

RID – Registered Identifier

GSC-IS OID – File ID or Object ID, 2 byte identifier of a particular container, as defined in the GSC-IS 2.1, not to be confused with a globally unique data object name in ASN.1 form (dot separated numeric values), the "OID" used by PIV end-point

PIX – 2-11 byte Proprietary Identifier extension

AID – Application Identifier

Universal AID – used to select generic containers or cryptographic modules, and referred to at the BSI level.

The RIDs of note are as follows:

A0 00 00 01 16	DoD PIV Transitional GSC-IS 2.1 data model, also PIV data model as specified by Table 1 in Section 1.7 of SP 800-73-1
A0 00 00 00 79	DoD – CAC data model. The CCC follows the GSC-IS 2.1 (and PIV) data model
A0 00 00 03 08	NIST – PIV end-point data model

From the BSI view in GSC-IS, PIV objects are referenced with a 7 byte **Universal AID** as follows:

In the middleware, this value is used to look up the Application Card URL in the CCC to retrieve the application ID (referred to as the PIX in SP 800-73-1) associated with this file. For CAC, the application ID and the object ID in the CCC are always the same, since each applet instance services a single container.

From the Card Edge view in GSC-IS and PIV, a SELECT command is issued to select applets and file objects. An applet selection data field contains a 5-16 byte identifier that can be a RID or a RID qualified by PIX.

An object within the selected application is referenced from the card edge by its GSC-IS Object ID (2 bytes).