DMDC

Card Technologies & Identity Solutions Division (CTIS)

DoD Implementation Guide for CAC PIV End-Point
Version 2.1.1

CTIS – Compliance & Standard Team

April 2010

U.S. Department of Defense (DoD)
Defense Manpower Data Center (DMDC)
Card Technologies & Identity Solutions Division (CTIS)
CTIS – Compliance & Standard Team
Dr. Robert Van Spyk
Marlon A. Guarino
Reed T. Anderson

Revision History

Version	Release Date	Description
v1.22	July 2008	Initial release. Based on SP 800-73-1 and FIPS201-1
v2.1	November 2009	Revised document includes: • Revised CAC PIV End-Point Data Model to 6.2.6b throughout document • Updated and added clarifications to various subjects. These include: the PIV transitional, Card Capability Container (CCC) structure implementation, the NACI, and Appendix E “Sample Java Program: Accessing the CHUID” • Added description of previous Applet suite 2.6.2 to Appendix I • Removed references to CACv1 • Added more details to PIV Authentication Certificate, Appendix C sample data, and clarified functionality at the card edge • Added descriptions for FASC-N elements • Re-organized sections for readability
V2.1.1	April 2010	Minor changes: • Added appendix J for RSA 2048 signing/decrypt • Added Index section • Editorial miscellaneous changes to Sect. 5.1 CCC

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. Technical details of PIV are captured in SP 800-73-1.

The Department of Defense (DoD) Common Access Card (CAC) ecosystem is compliant with SP 800-73-1. This version of the Implementation Guide brings CAC and PIV End-Point up-to-date with current, issued DoD CAC PIV End-Point cards.

The CAC/PIV world evolves constantly. Although this guidance is based strictly on SP 800-73-1, it recognizes the editorial contribution of SP 800-73-2, the ongoing cryptographic migration specified in SP 800-78-1, and the introduction of 128K cards. Physical access and contactless standards are also evolving rapidly but will be mentioned here only in-so-far as they are relevant to existing implementation.

The scope of this Guide is the CAC and PIV End-Point. The Transitional PIV[1] is no longer relevant to the development audience addressed here.

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 specific details of the PIV End-Point as implemented on the CAC PIV End-Point with Transitional backwards compatibility. Middleware specifications are discussed in other documents.

1.3 Scope & Audience

The Scope of this document is the DoD CAC PIV End-Point with PIV End-Point interfaces and the data model as described in SP 800-73-1 and implemented within the CAC 2.6.2b applet suite[2].

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

- The PIV and CAC applications have clearly defined dependencies.
- CAC is the primary application for DoD
- The CAC is NIST Interagency Report 6887 GSC-IS 2.1 compliant [GSC-IS]
- DoD adds PIV mandatory data model and card edge elements to the CAC

[1] For the sake of simplicity the term “CAC Next Generation” or “NG” is also referred to as the “PIV Transitional” or “Transitional” in line with NIST SP 800-73-1 part 2.
[2] Previous CAC PIV Transitional and End-Point platforms implemented the CAC 2.6.2 applet suite outlined in appendix K. The only difference is the process and means to request/retrieve the data.

- This guide references the Java Card 2.x implementation for illustrative purposes; guidance is language independent
- Optional SP 800-73-1 elements may be mandated for internal use by DoD
- Backward compatibility with existing middleware and card is maintained
- DoD middleware has the ability to communicate with both CAC, 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 implemented on the DoD CAC PIV End-Point, 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. This dates back to 1999, when 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, signature, and encryption certificates. Externally, it contains a linear barcode, two-dimensional barcode, magnetic stripe, color digital photograph, and printed text [3].

CAC PIV End-Point cards currently support RSA cryptographic algorithms of either RSA-1024 or RSA-2048 with SHA-1 (SHA-256 in the near future) for key generation. Most cards currently in use support RSA-1024 with SHA-1. We have begun issuing cards supporting RSA-2048. SP 800-78 specifies the time line for the cryptographic evolution to improved algorithms and larger key sizes.

2.1 Interoperability

The CAC PIV End-Point has an additional applet with PIV End-Point functionality. Within the DoD, the CAC functions as the primary interoperable credential[4]. Outside the DoD; the PIV End-Point is the primary interoperable credential.

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

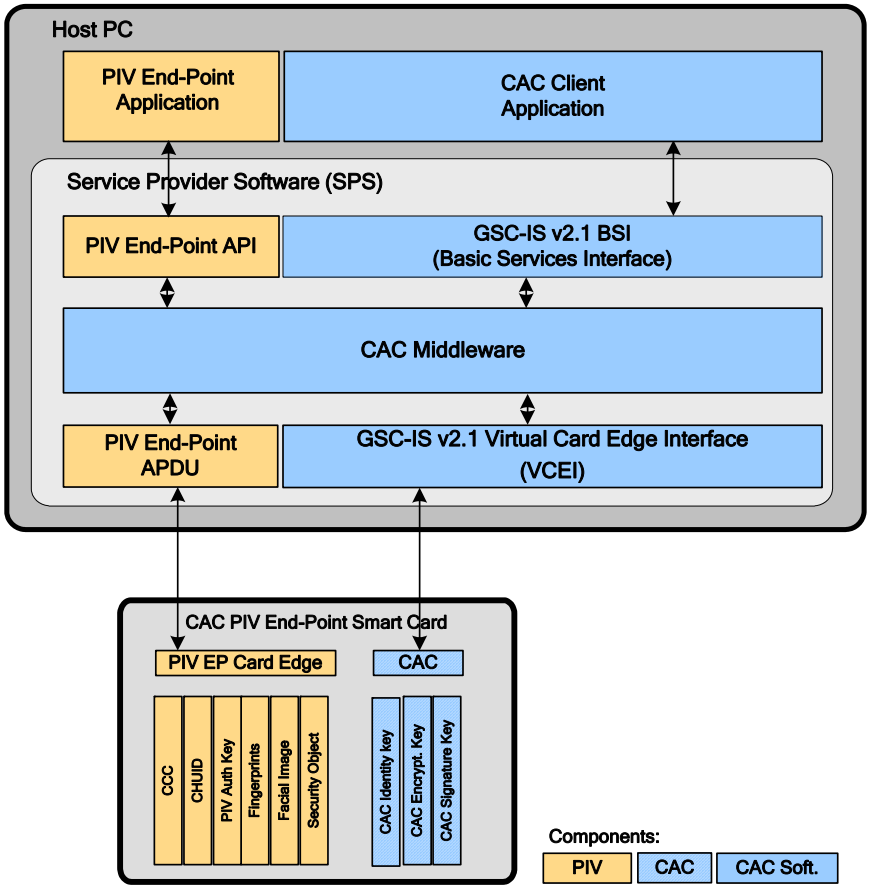
[4] Unique to the DoD and for the purpose of backward compatibility, both PIV Transitional and End-Point can surface at the card edge. Although this capability is transparent for the most part, it is relevant only for so long as this type of card is in circulation.

2.2 PIV and CAC Components

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

A PIV host application will use the PIV on-card application for physical or logical access, communicating via the SP 800-73-1 API interfaces for the PIV End-Point.

Figure 1. Sample CAC and PIV components



3 DATA MODEL DISCOVERY

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. Table 1 below summarizes the data models that can be expected.

Table 1. Card Data Models

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

The data model number for CAC may only range from 00-04.[5]

3.1 Discovery for Contact Card

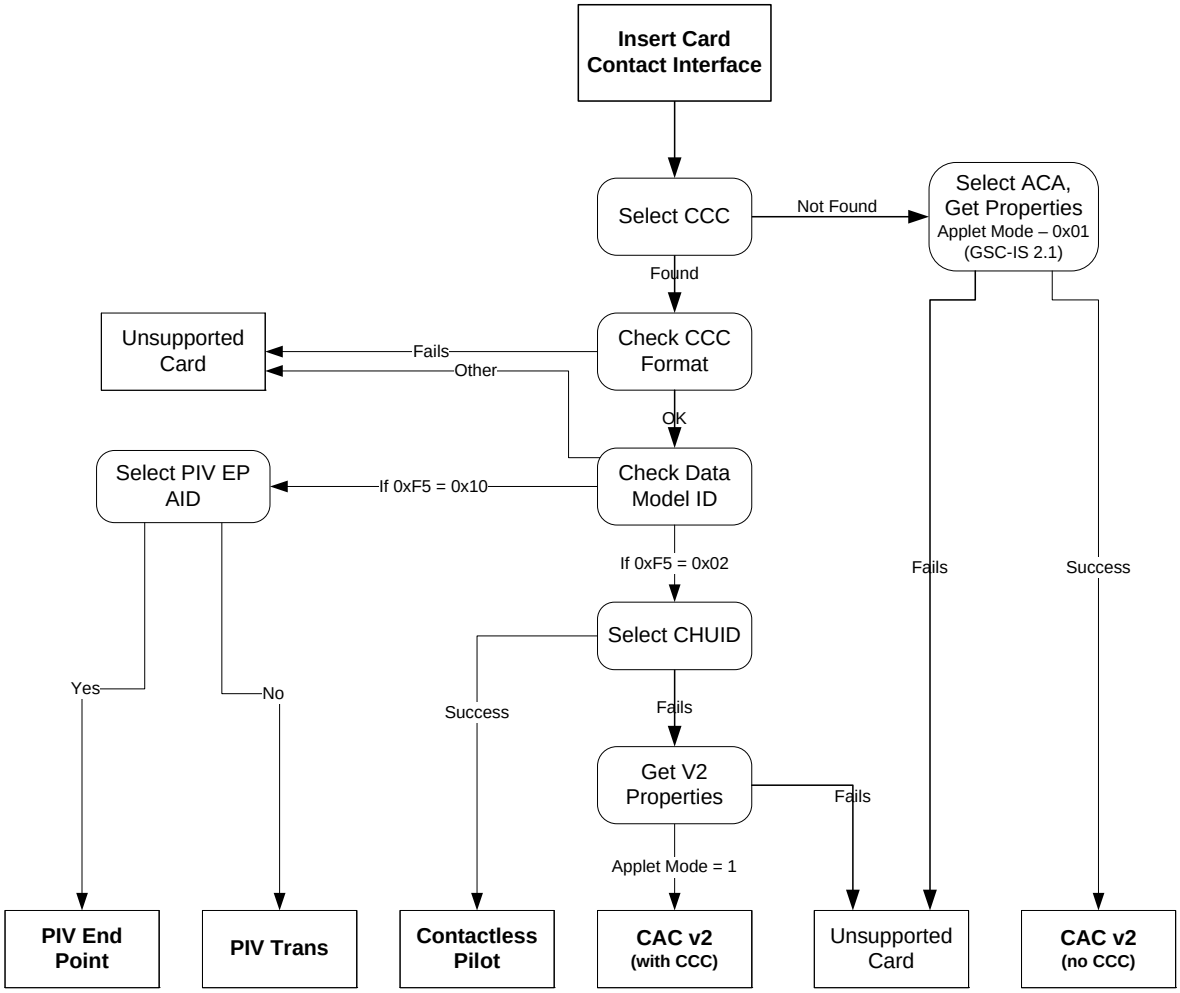
The criteria used to discover the card data model are:

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

Figure 2 depicts one possible method[6] of middleware discovery for contact cards supporting CACv2, PIV Transitional, or CAC PIV End-Point.

[5] Standard GSC-IS profiles used 0x01 data model number.
[6] There may be other methods of discovery.

Figure 2. Contact Card Discovery Flow



Note: The CHUID is read in the clear over the contact interface.

3.2 Discovery for Contactless Card

Contactless discovery of the PIV End-Point is based on the selection of the default container. The default container is the CHUID. It is read in the clear over the contactless interface. Efforts are underway to protect this interface cryptographically so that both the credential and the reader can be trusted.

The minimum interoperability mechanism for cardholder identification is to read the CHUID from a fixed location using READ BINARY and SELECT EF ISO 7816-4 [ISO4]. GET DATA for contactless is an End-Point command. Physical access device vendors may support both READ BINARY and GET DATA for interoperability.

3.3 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 readers conform to ISO 14443 Parts 1 through 4. Cryptographic functionality is not required.

3.4 Default Selected Container

Table 2 below summarizes default selected containers depending on card type issued and mode in use.

Table 2. Default Selected Containers

Card Type	Contact	Contactless
CACv2	CardManager	N/A
PIV Transitional	CHUID (Transitional)	CHUID (Transitional)
PIV End-Point card	CHUID	CHUID

4 CAC PIV End-Point Consolidated Data Model

This section contains the Consolidated DoD PIV End-Point (EP) data model (Figure 3) for use with the 2.6.2B applet (for the non-consolidated data model, see Appendix I). 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 PIV 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 Authentication certificate [7]

The functionality of these optional PIV certificates (containers) is available through the existing CAC certificates:

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

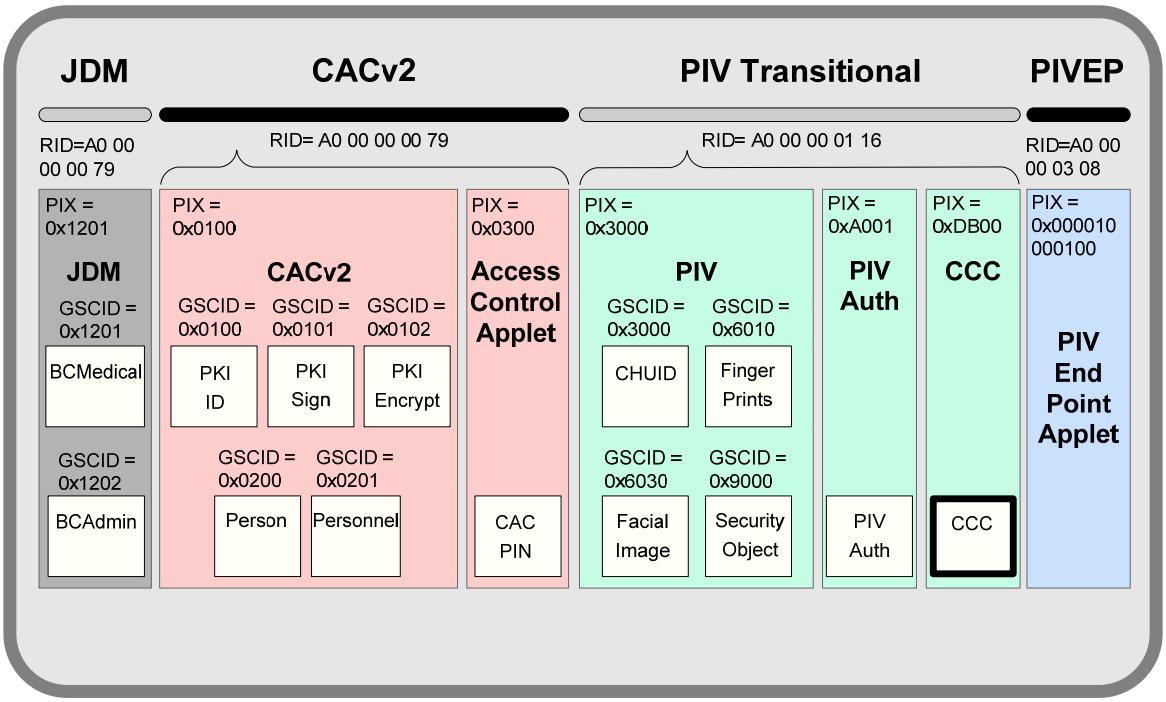
The remaining optional PIV containers are not implemented:

- Card Authentication Key
- Printed Information Buffer

The JDM container shown below is not part of the PIV data model and out of scope for this guidance, however it is shown here for completeness on what is stored on card.

[7] See Appendix D Sample data for PIV Authentication certificate.

Figure 3. CAC PIV End-Point Consolidated Data Model



SP 800-73-1 permits default applets[8]. 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. PIV Transitional and PIV End-Point data elements/containers are shared. The End-Point accesses the secure data through the PIV End-Point card edge[9], with differences noted throughout this document.

4.1 Access Control Rules

Table 3 lists the Access Control Rules supported by PIV End-Point applet during the "Card Usage" phase of the card lifecycle[10].

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 once
PIN Always / PIN ALW	The corresponding service can be provided if the PIV application PIN code value has been verified for each operation.

[8] SP 800-73-1 section 3.4.2, pg 19 does not mandate a specific selected application.
[9] The legacy Transitional may be accessed through the CAC GSC-IS card edge.
[10] These are not applicable at the "Card Issuance" phase of the lifecycle.

Table 4 lists all the BER-TLV tags supported by the PIV End-Point applet, the associated key reference (when applicable), and the access control rules associated with each of them. Optional objects not implemented by DoD are grayed out.

Table 4. PIV Data Objects Access Control Rules

Data Object	Tag[11]	OID[12]	Key	Contact			Contactless			M / O
	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	0101[13]	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[14]	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
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 either "GENERAL AUTHENTICATE" operations or PIN verification.
- "**M/O**" stands for Mandatory or Optional
- "**N/A**" stands for Not Applicable

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

[12] OID is not interpreted at the applet level. The link between BER-TLV tags and OIDs is described here for reference only

[13] The PIV Authentication certificate OID is 0x0101 if accessed through the End-Point card edge.

[14] PIN if accessed via the Transitional interface.

5 DoD PIV End-Point Data Elements

The CAC PIV data model containers are further defined in the following sections.[15]

5.1 CCC (0xDB00)

The Card Capabilities Container (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 DoD container elements of the CCC are defined in Table 5.

Table 5. CCC Container Elements

Element	Tag	Len	Value(s)	Comments
Card Identifier	0xF0	0x15	GSC-RID (5B) manufacturer-Id (1B) Card-type (1B) Card-Id (14B)	See Card Identifier Note below. Card-Id = CUID (10B) + Manufacturer batch serial (4B)
Capability Version	0xF1	0x01	0x21	Meaning GSC-IS v2.1
Capability Grammar Version	0xF2	0x01	0x21	Meaning GSC-IS v2.1
ApplicationsCardURL	0xF3	0x10	RID (5B) AppType (1B) ObjectID (2B) AppID (2B) AccProfile (1B) PinID (1B) AccKeyInfo (4B)	e.g. A000000079, A000000116. See Table 6 for actual values on card. 0x01 = GC; 0x04 = PKI; 0x00 = PIV; 0xA0 for 3rd party applets e.g. 0x0200 Person Object. e.g AppID for CAC 0x0100 not used for Java Card, 00 not used for Java Card, 00 not used for Java Card, 00000000
PKCS#15	0xF4	0x01	0x00	Meaning not PKCS#15 token
Registered Data Model	0xF5	0x01	0x10[16]	
Access Control Rule Table	0xF6	0x11	0x07 0xA0 0x00 0x00 0x00 0x79 0x03 0x00 0000000000000000	Choice of acrTableAID
Card APDUs	0xF7	0x00		(not used)
Redirection Tag	0xFA	0x00		(not used)
Capability Tuples	0xFB	0x00		(not used)
Status Tuples	0xFC	0x00		(not used)
Next CCC	0xFD	0x00		(not used)
Error Detection Code	0xFE	0x00		(not used)

(**Card Identifier Note:** 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.)

[15] All PIV End-Point Data Elements contain an error detection code tag, however no values are assigned.

[16] Implementers should be aware that data model numbers may change as revisions to SP 800-73 occur.

5.1.1 ApplicationsCardURL

This section details the format of the PIV CCC ApplicationsCardURL element, combining the specifications from SP 800-73-1 and GSC-IS 2.1. Table 6 below lists sample application URL entries used by the DoD. Current GSC-IS and PIV CCC URL entries are 18 bytes each (tag 1 byte + length 1 byte + value 16 bytes) beginning with the tag 0xF3, denoting a GSC-IS v2.1 compliant object.

Table 6. Sample PIV End-Point ApplicationCardURL Values

RID	App ID	Object ID	Application	Applications Card URL
A000000116	3000	3000	Card Holder Unique Identifier (CHUID)	F3 10 A000000116 01 3000 3000 00 00 00000000
A000000116	3000	6010	Card Holder Fingerprints Container	F3 10 A000000116 01 3000 6010 00 00 00000000
A000000116	3000	9000	Security Object	F3 10 A000000116 04 3000 9000 00 00 00000000
A000000116	3000	6030	Facial Image	F3 10 A000000116 01 3000 6030 00 00 00000000
…	…	…	…	…

Note: SP 800-73-1 specifies the RID of A0000000308 for the CCC, however DoD uses the RID A0000000116 for backwards compatibility.

Figure 4 depicts the GSC-IS card ApplicationCardURL format for the CAC/PIV CCC.

Figure 4. GSC-IS CCC ApplicationCardURL Encoding

5.1.2 CCC Usage

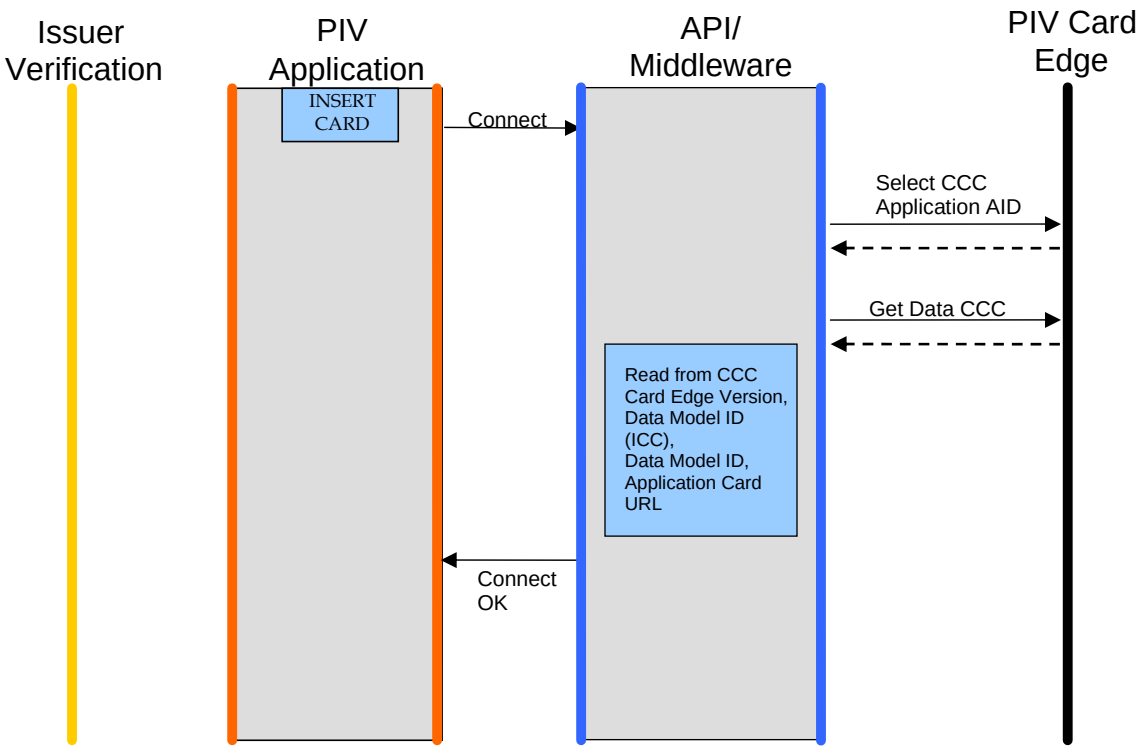
The CCC container is used to allow fast and light discovery of card capabilities. Middleware implementers should read relevant information from the CCC, rather than hard coding values in the middleware for specific cards. The CCC provides information allowing:

- Confirmation that the data model ID is known, and therefore, the data model scope and format can be handled by the middleware.
- Discovery of 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 is located in the ApplicationsCardURL attributes.

Retrieval of the CCC is depicted in Figure 5 below.

Figure 5. Retrieval of CCC Sequence Diagram

(It is strongly recommended that this process be used before any other transaction may begin)



5.1.2.1 CCC Access Pseudo Code

This section contains pseudo code for accessing the CCC, and sample contents of the CCC in a CACv2 and PIV Container.

1. SELECT End-Point AID
2. GET DATA '5FC107' (CCC)

CCC access condition: Read ALW; Update OP-SC

CCC container should contain TLV items from Table 5 above.

5.1.3 CCC Discrepancies

The CCC content conforms to GSC-IS v2.1 (NIST 6887). The discrepancies between GSC-IS and SP 800-73-1 are compared in Table 7 below:

Table 7. CCC Comparison (GSC-IS v2.1 vs. SP 800-73-1)

CCC Data Element	Tag	Attribute	(GSC-IS v2.1)	SP 800-73-1
CardIdentifier	F0	Length	Variable	21 fixed
ApplicationCardURL	F3	Length	Variable	Variable limited to 128 bytes
	F3	RID	A0.00.00.01.16	A0.00.00.03.08
Registered Data model	F5	Value	0x01	0x10
Access Control Rule Table	F6	Length	Variable	17 bytes
Card APDUs	F7	Length	6 bytes	0 byte
Redirection Tag	FA	Length	Variable	0 byte
Capability Tuples	FB	Length	Variable	0 byte
Status Tuples	FC	Length	Variable	0 byte
next CCC	FD	Length	Variable	0 byte

DoD implements the PIV CCC which is limited to 266 bytes in length (as per SP 800-73-1). This differs from the GSC-IS v2.1 CCC which is variable in length.

According to NIST, these discrepancies are acceptable as long as the CCC content remains compliant with GSC-IS v2.1 specifications. By definition, PIV EP middleware will accept the listed discrepancies (assuming such PIV middleware is reading the CCC).

5.2 CHUID (0x3000)

Further clarifications of CHUID fields 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. Fields not implemented are grayed out on Table 8 below.

Table 8. 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 FASC-N is not allowed to be modified post-issuance.
- GUID is reserved for future use. GUID field is expected to be 16 bytes of binary 0x00.
- In machine readable format, the expiration date element specifies the expiration date. The format and encoding are specified in SP 800-73-1. This date is identical to the one printed on the card surface. The optional Printed Information Buffer is not included.

- 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].
    - Includes the Issuing Authority X.509 Signature Certificate in the container.

5.2.1 CHUID Usage

As outlined in the NIST Special Publication 800-73-1 [SP800-73] the CHUID is defined as follows:

- 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].
- The PIV CHUID is a free read from both the contact and contactless interfaces of the PIV Card
- Algorithm and key size requirements for the Issuer asymmetric signature are detailed in NIST SP 800-78 [SP800-78].
- 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 in length

See Appendix E for a code sample on retrieving the CHUID.

5.2.2 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” as part of the Credential Number, increases the limit of cards issued per site from 999,999, to 99,999,999 to accommodate a larger population.

The FASC-N is defined in the Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems, Version 2.3E [PACS 2.3E]. It consists of 40 total characters encoded as Packed Binary Coded Decimal (BCD) format with odd parity creating a 200 bit (25 byte) record.

Table 9 below details the fields of the FASC-N, their length and their description. The last 16 characters uniquely identify the card holder.

Table 9. FASC-N

	Field Name	Length (BCD Digits)	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	Extends the Credential Number
	Individual Credential Issue	1	Extends the Credential Number
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.

5.2.2.1 Agency Code

The Agency Code (AC) FASC-N data element identifies the government agency that is issuing the credential. The definitions that are used for this field are identified in SP 800-87. The codes defined there represent the congressional code for budget execution or payment items in reporting to Office of Management and Budget [OMB]. The Agency Code will be determined by the agency affiliation of the site issuing the Common Access Card. Table 10 outlines the possible Agency Code values based on the site's Affiliation The DoD has no alphanumeric AC.

Table 10 Agency Code

Affiliation	Agency Code
Department of the Army	2100
Department of the Navy	1700
Department of the Navy – U.S. Marine Corps	1727
Department of the Air Force	5700
Department of Defense – Other Agencies	9700
U.S. Coast Guard	7008
U.S. Public Health Service	7520
National Oceanic and Atmospheric Administration	1330

5.2.2.2 System Code

The System Code (SC) identifies the system the card is enrolled in and is unique for each site. Number assignment is the responsibility of the CIO of the organization referenced by

the Agency Code. The DoD has chosen to use this field to identify the site that issued the card.

5.2.2.3 Credential Number

Used in the combination of an Agency Code, System Code, Credential Number, credential series, and Individual Credential Issue to create a fully qualified number that uniquely defines the card.

5.2.2.4 Person Identifier

DoD Person Identifier (PI) is Electronic Data Interchange Person Identifier (EDIPI). This is the unique number within DoD to identify an individual.

5.2.2.5 Organizational Category

The Organizational Category (OC) is used to indicate what is being used as an organizational Identifier. The optional values for this field defined in Table 11:

Table 11 Organizational Category code

Organization	OC Code
Federal Government Agency	1
State Government Agency	2
Commercial Enterprise	3
Foreign Government	4
Locally Assigned	5

The DoD will use 1 or 5 in this field.

5.2.2.6 Organizational Identifier

The Organizational Identifier (OI) indicates the organization the card recipient is employed or sponsored by and where the person's identity and association information can be accessed for authentication of card and cardholder. The values for this field are:

If Organizational Category is 1 then Organizational Identifier is the SP 800-87 Organization Code.
If Organizational Category is 2 then Organizational Identifier is the State Code.
If Organizational Category is 3 then Organizational Identifier is the Company Code.
If Organizational Category is 4 then Organizational Identifier is the Country Code.

The DoD will use the SP 800-87 Organization Code of the employing/sponsoring agency.

5.2.2.7 Person/Organization Association Category

The Person/Organization Association Category (POA) identifies the association of the card holder to the employing/sponsoring agency. The values for this field are defined in Table 12:

Table 12 Person/Organization Association Category

Person/Organization Association	POA Code
Employee	1
Civil	2
Executive Staff	3
Uniformed Service	4
Contractor	5
Organization Affiliate	6
Organization Beneficiary	7

5.2.3 GUID

The GUID is reserved for future use.

5.2.4 Issuer Asymmetric Signature

The DoD Issuer Asymmetric Digital Signature is used to sign several objects on the card. The DoD retrieves the digital certificate from a Defense Information Systems Agency (DISA) approved Certificate Authority.

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

1. A binary string representing the plain-text concatenation of the FASC-N, GUID and Expiration Date.
2. The hash is then computed on this plain-text string by the card issuer, using the digestAlgorithm specified in the SignedData object.
3. The resulting hash is signed by the card issuer, using the `signatureAlgorithm` specified in `SignedData` object defined below.

Note: Neither this signature nor the hash are part of the PIV Security Object container defined later in this section. It is part of the asymmetric signature field's `SignedData` object found in the CHUID. Table 13 details the contents of the SignedData Object.

Table 13. Issuer Asymmetric Signature "SignedData" Object

Value	m / o	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.
signingTime	o	May be included in RSA 2048 certificates.
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 PIV Authentication Certificate (0x0101)

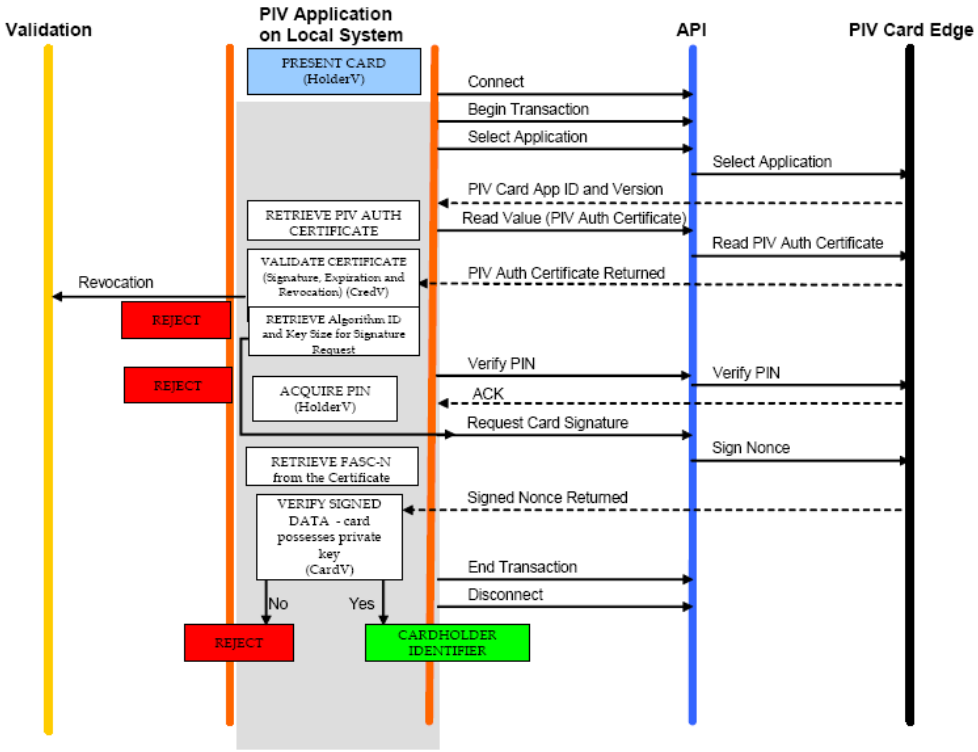
This certificate is used to authenticate the card holder for logical access. The CAC PKI Signature key and associated certificate is used for Microsoft cryptographic logon and PKI signature. The DoD certificate does not include the NACI (as specified by FIPS 201), but it does contain the FASC-N.

The UPN field is populated with the email address created from the last 16 characters of the FASCN-N and the email domain (i.e. 1111111111111111@mil)[17].

To access the PIV Authentication certificate from the End-Point card edge, implementers first select the PIV EP applet and then issue a GET_DATA request as per SP 800-73-1. To access the PIV Authentication certificate from the PIV Transitional card edge, implementers use OID 0xA001.

Figure 6 below illustrates the process of authenticating the PIV Authentication certificate.

Figure 6. Authenticate PIV Authentication Key Sequence diagram



5.3.1 PIV Authentication Certificate in the CAC Environment

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

Table 14 exhibits the access control rules for PIV and CAC certificate usage. The keys listed under "*SP-800-73-1 Certificates*" are the optional and mandatory certificate (keys) proposed by the PIV standard. The keys listed under "*CAC Certificates*" are used by the DoD today.

[17] The UPN in the PIV Authentication certificate is different from the UPN in the PKI Signing certificate, which is EDI@mil.

In the Transitional PIV implementation for the DoD, the PIV Authentication certificate equates to the DoD PKI Signature certificate for cryptographic logon to a Microsoft operating system, or to the PKI Identity certificate for establishing secure connections to websites.

All End-Point keys issued today are RSA-2048, completing part of the SP 800-78-1 mandated cryptographic improvement. Thus, the entire trust chain now utilizes RSA-2048 bit keys. Further changes will include migration to SHA-256, AES, and ECC.

Table 14. PIV, CAC Key, and Certificate Access Rules

	Key Name	Key Purpose	Access Read / Usage	OID	M/O
PIV Certificates	PIV Authentication Certificate	Used to Authenticate the card and the Card Holder using PIN. Identity key for logical access.	ALW/PIN	0x0101*	M
	Digital Signature Certificate	Digital Signature for non-repudiation. Contact only	ALW/PIN-Always	0x0100	O
	Key Management Certificate	Encryption key for confidentiality. Contact only	ALW/PIN	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 (Email encryption)	ALW/PIN	0x0102	M
	Note: The gray area in the table indicates optional PIV keys which the DoD functionally implemented using the existing DoD PKI keys. ***** When the PIV Authentication certificate is accessed via the Transitional card edge with the PIV Authentication certificate activated, the OID is 0xA001.				

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

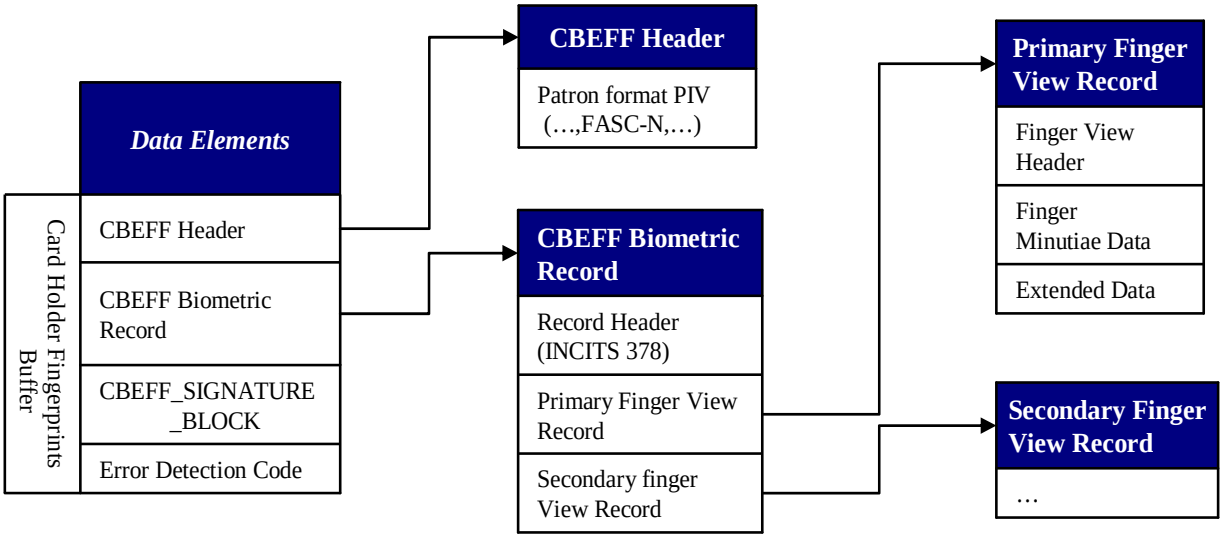
5.4 Card Holder Fingerprints Container (0x6010)

The Card Holder Fingerprints Container is mandatory and is implemented in accordance to SP 800-73-1.

Note: The following finger minutiae changes were specified with SP 800-73-1:

- 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 SP 800-76 table 3 and INCITS 378 Sec 6 table 6.

5.4.1 CBEFF Biometric Record Overview

In accordance with SP 800-76, fingerprint minutiae and facial image are stored on the card. These are also stored in the 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) [378]. This specifies that all biometric data shall be digitally signed and uniformly encapsulated. This covers the PIV Card fingerprints mandated by FIPS 201 and the Facial Image.

All such data is signed in the same manner as prescribed in FIPS 201 and SP 800-73 for the biometric elements. The issuer signature is present for integrity and is stored in the CBEFF signature block; the issuer signature certificate is in the CHUID. The overall arrangement of CBEFF and references is depicted in Table 15 below:

Table 15. Simple CBEFF Structure

Data Element	References
CBEFF_HEADER	One instance of the CBEFF header (SP 800-76 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 (SP 800-76 Sec 6 Table 7), other references: FIPS 201 4.4.2, INCITS 398 5.2.3. CMS-compliant Issuer signature including FASC-N.

The SP 800-76 template specification restricts the options of INCITS 378;

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

5.4.1.1 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 F *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).

5.4.1.2 CBEFF Signature Block

Details on the process for generating and populating the CBEFF Signature Block are described in FIPS 201 sec 4.4.2 and SP 800-73. 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 PIV 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. It includes a CBEFF signature block with the FASC-N as a signed attribute and is protected by the security object.

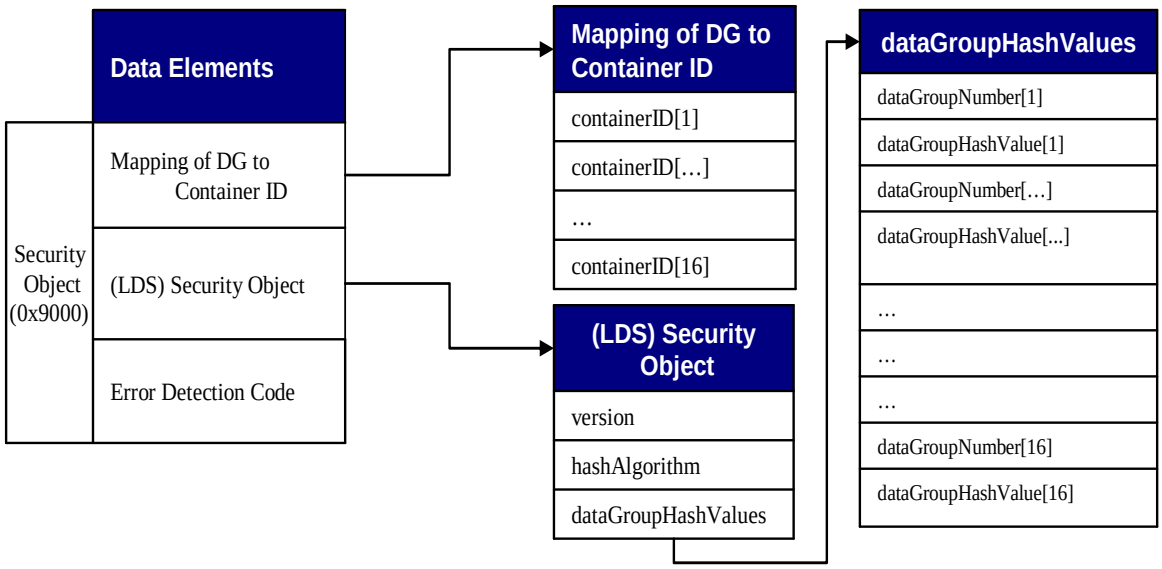
5.6 Security Object (0x9000)

The principal goal of the security object is to reduce the number of cryptographic operations performed on data objects when verify their issuer signature. It provides a means for verifying the integrity of card data elements that bind a card to the card holder's identity with minimal processing. The security Object container (0x9000) is signed by the issuer; however, the issuer's certificate is not included with the Security Object, since it is already part of the CHUID. It follows the Machine Readable Travel Document (MRTD) format and includes the tag (0xBA) for mapping the PIV containerIDs to the MRTD Data Groups. For containers not present on the card, the container IDs mapped to their respective Data Groups are null.

The LDS Security Object itself (0xBB) contains (at a minimum) unsigned data objects, such as the Printed Information data object. For maximum protection against credential splicing attacks (credential substitution), it is recommended, however, that all PIV data objects, except the PIV X.509 certificates, be included in the Security Object [SP800-73]

The LDS Security Object also contains hashes of the mapped containers. The hashes include the entire contents – even the signature. They appear in the order in which data elements are presented in the PIV data model overview, but the order is not important.

Figure 8. Security Object Structure



Note: In SP 800-73-1, the data structure for "Mapping of DG to Container ID" is not defined.

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

Table 16. 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 17. 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

5.6.1 Mapping of Data Groups to PIV Containers

DoD implementation uses explicit mapping and populates each entry into the table with a 1 byte index followed by a 2 byte container ID. The previous data model instantiated all 16 entries. See Appendix I for further details.

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 18. Mapping of Data Groups to Container ID

DataGroup Number	container ID	dataGroup HashValue	References to SP 800-73-1	Comment
2	0x6030	MIT	Image for Visual Verification (to be consistent with previous references in this document)	Not the same as in the LDS Encoded Face.
3	0x6010	MIT	Card Holder Fingerprints (Appendix F)	Similar to LDS Encode Finger Datagroup [MRTD].

(MIT) Mandatory at time of instantiation.

Note: Data groups not implemented are not shown.

6 CONFORMANCE TESTING

6.1 CAC End-Point implementation conformance Testing

DoD Complies with SP 800-85b testing except in the case of one data element:

- NACI OID for the NACI indicator extension which tells the status of the subject's background investigation at the time of credential issuance.

ICAM membership agreed to exempt DoD from this requirement as a legacy smart card implementation. DoD uses Backend Attribute Exchange (BAE) services as an alternative.

Appendix A Definitions & 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
JDM	Joint Data Model
NAC	National Agency Check
OID	Object ID
PDR	Personal Data Repository

PIV	Personal Identity Verification
PIX	Proprietary Identifier Extension
PKI	Public Key Infrastructure
RID	Registered Identifier
TLV	Tag Length Value

Appendix B References

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

[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 SP 800-73-1*, 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 800-85B, *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.

[TB] DMDC Technical Bulletin: *CAC Data Model Change in 144k Dual Interface Cards*, February 13, 2009

Appendix C Sample data for PIV Authentication Certificate

Below is a sample record format for the DoD RSA-2048 PIV Authentication certificate. This End-Entity was issued by a RSA-2048 Root CA and a RSA-2048 Intermediate CA. 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
Comments are enclosed in parenthesis to clarify raw data.

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 2048 bit 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 2048 bit 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**)	(OID Indicator for **SC Logon**)
		2B 06 01 05 05 07 03 02 (1.3.6.1.5.5.7.3.2= **id_kp_clientAuth**)	(OID For SSL authentication)
*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**)	(OID for cert policies per RFC 3280)
		60 86 48 01 65 02 01 0B 09 (2.16.840.1.101.2.1.11.9 =null)	(OID for DoD Medium Hardware Assurance)
*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**)	(OID for **UPN**; value below is used for SC Logon)
		31 34 30 31 33 35 37 33 35 38 31 35 37 30 30 34 40 6D 69 6C	'1401357358157004@mil' (UPN created from FASC-N (OC+OE+PI+POA+@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_authorityInfoAccess**)	
		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**)	(Certificate Revocation List URL)
		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**))	(Key usage follows)
		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)	SHA1 withRSA 2048 bit 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 D DoD CAC PIV End-Point Quick Guide

DoD CAC PIV End-Point SP 800-73-1

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	pki – Compute 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

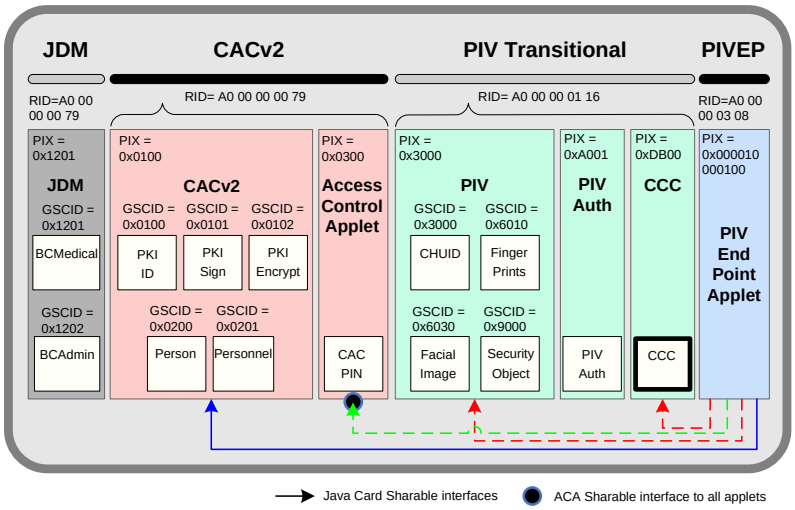
Security Object		0x9000	Always Read
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Mapping of DG to ContainerID	0xBA	Variable	100
Security Object (Issuer Signature)	0xBB	Variable	900
Error Detection Code	0xFE	LRC	0

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 (36 BCD Digits)			
Field Name	L (BCD)	Field description	
AGENCY CODE	4	Identifies the government agency issuing the credential	Concatenated 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 – Country Code OC=4 – Numeric Country Code	
POA	1	PERSON/ORGANIZATION ASSOCIATION CATEGORY 1 – Employee 2 – Civil 3 – Executive Staff 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



DMDC CTIS/Technical Advisory Group (TAG) 9-1-09

Appendix E Sample Java Program: Accessing the CHUID

Below, is a sample Java program accessing the CHUID using NIST SP 800-73-1 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/DMDC, cac.lab@osd.pentagon.mil
 *
 * DISCLAIMER: This software is released by the CAC Test Lab as a serv-
 * ice and is expressly provided "AS IS." This software was developed
 * for demonstrative or educational purposes. The author and CAC Test
 * Lab/DMDC MAKE NO WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY
 * , INDCLUDING, WITHOUT LIMITATION, THE IMPLIED WARANTY OF MERCHANTAB-
 * ILITY, FITNESS FOR A PARTICULAR PURPOSE, NON INFRINGEMENT AND DATA
 * ACCURACY. THE AUTHOR AND CAC TEST LAB/DMDC DO NOT REPRESENT OR WARR-
 * ANT THAT THE OPERATION OF THIS SOFTWARE WILL BE UNINTERRUPTED OR ER-
 * ROR-FREE, OR THAT ANY DEFECTS WILL BE CORRECTED.
 *
 * Required JDK 1.6 and above to compile and run.
 */

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

 public class GetChuidPivEp{
    public static SmartCardApdus apdus = new SmartCardApdus();
    public static ApduUtilities utils = new ApduUtilities();

    /**
     * Main method
     */
    public static void main(String[] args) throws CardException,
UnsupportedEncodingException {
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        System.out.println("          ------------------------------" );
        System.out.println("          ====Java Smartcard IO Test====");

        try{
            smartIoApiEx();
        }catch(CardException e){
            System.out.println("Error occured in transaction: "+e);
        }
        System.out.println();
        System.out.println("press <Enter> to quit");
        try{in.readLine();}catch(IOException e){System.out.println("IO
error");}
        return;
    }

    /**
```

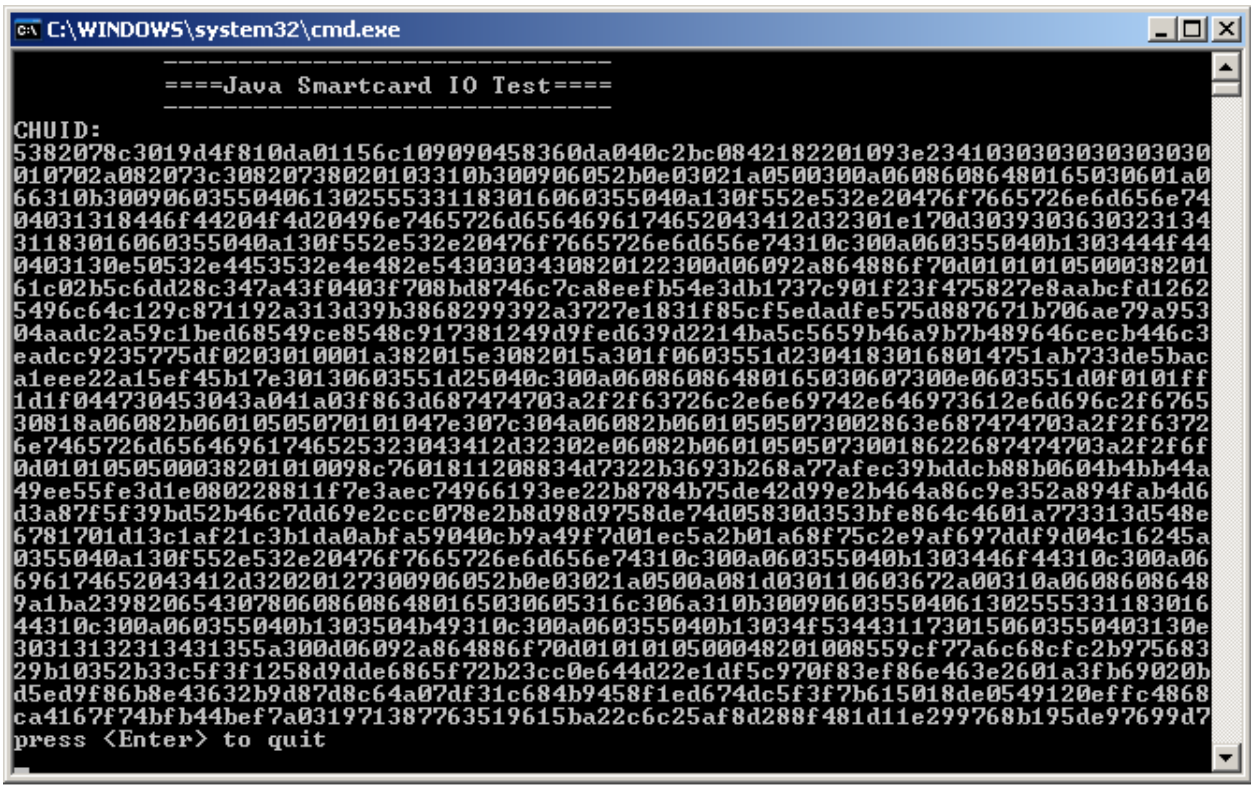
```
     * Java (smartcardio) smart card connection/communication setup
     */
    public static void smartIoApiEx() throws CardException,
UnsupportedEncodingException {
        //the AID for PIV Application
        int termNum = 0;

        CardConnection cConnection = new CardConnection();
        CardChannel channel = cConnection.openCardChannel(termNum, "*");

        apdus.select(channel, utils.PIV_EP_APP_AID);
        byte[] chuid = apdus.getData(channel, utils.PIV_EP_APP_CHUID_OID);

        //output data (e.g. console, file, etc.)
        System.out.println("          ----------------------------" );
        System.out.println("CHUID: ");
        System.out.print(utils.binToHexString(chuid));

        //disconnect
        cConnection.closeCardChannel();
        return;
    }
}

/*
 * smartcardio test program. DOD/PIV related APDU helper functions.
 * @author J. Arana & N. Kubiak, CAC Test Lab/DMDC,
 * cac.lab@osd.pentagon.mil
 *
 * DISCLAIMER: This software is released by the CAC Test Lab as a serv-
 * ice and is expressly provided "AS IS." This software was developed
 * for demonstrative or educational purposes. The author and CAC Test
 * Lab/DMDC MAKE NO WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY
 * , INDCLUDING, WITHOUT LIMITATION, THE IMPLIED WARANTY OF MERCHANTAB-
 * ILITY, FITNESS FOR A PARTICULAR PURPOSE, NON INFRINGEMENT AND DATA
 * ACCURACY. THE AUTHOR AND CAC TEST LAB/DMDC DO NOT REPRESENT OR WARR-
 * ANT THAT THE OPERATION OF THIS SOFTWARE WILL BE UNINTERRUPTED OR ER-
 * ROR-FREE, OR THAT ANY DEFECTS WILL BE CORRECTED.
 */
import java.io.*;
import javax.smartcardio.*;

public class SmartCardApdus{
    private static boolean printApdus = false;
    private static boolean printExceptions = false;
    private static final ApduUtilities utils = new ApduUtilities();

    /*
     * select an applet or container in a smart card
     */
    public static ResponseAPDU select(CardChannel channel, byte[] data)
throws CardException, UnsupportedEncodingException{
        byte cla = 0x00,
             ins = (byte)(0xA4 & 0xFF),
              p1 = 0x04,
              p2 = 0x00,
              de = 0x00;
```

```
        int  sw1 = 0,
             sw2 = 0;
        CommandAPDU cmd = null;
        ResponseAPDU res = null;
        try {
            cmd = new CommandAPDU(cla, ins, p1, p2, data, de);
            if(printApdus) {
                System.out.println("CommandAPDU:
"+utils.binToHexString(cmd.getBytes()));
                System.out.println();
            }
            res = channel.transmit(cmd);
            if(printApdus) {
                System.out.println("ResponseAPDU:
"+utils.binToHexString(res.getBytes()));
                System.out.println();
            }
            if(res.getSW1() == 0x6D && res.getSW2() == 0x00)
                throw new CardException("SmartCardApdus: select(): Unknown
instruction given in the command");

        } catch (CardException e) {
            if(printExceptions) {
                System.out.println(e);
                System.out.println();
            }

        }
        return res;
    }

    /*
     * obtain the data of the currently selected app
     */
    public static byte[] getData(CardChannel channel, byte[] data) throws
CardException, UnsupportedEncodingException {
       int cla = 0x00, //the bytes in the APDU
           ins = (byte)0xCB,
            p1 = 0x3F,
            p2 = (byte)0xFF;
       byte[] data2 = new byte[5];
       data2[0]=0x5C;
       data2[1]=0x03;
       System.arraycopy(data, 0, data2, 2, 3);

       ResponseAPDU res = null;
       CommandAPDU cmd = null;

       cmd = new CommandAPDU(cla, ins, p1, p2, data2);
       if(printApdus) {
           System.out.println("CommandAPDU:
"+utils.binToHexString(cmd.getBytes()));
           System.out.println();
       }
       res = channel.transmit(cmd);
       if(printApdus) {
```

```
            System.out.println("ResponseAPDU:
"+utils.binToHexString(res.getBytes()));
            System.out.println();
        }
        return res.getBytes();
    }
}


 /*
 * smartcardio program. Misc functions that are helpful when dealing
 * with APDUs
 * @author J. Arana & N. Kubiak, CAC Test Lab/DMDC,
 * cac.lab@osd.pentagon.mil
 *
 * DISCLAIMER: This software is released by the CAC Test Lab as a serv-
 * ice and is expressly provided "AS IS." This software was developed
 * for demonstrative or educational purposes. The author and CAC Test
 * Lab/DMDC MAKE NO WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY
 * , INDCLUDING, WITHOUT LIMITATION, THE IMPLIED WARANTY OF MERCHANTAB-
 * ILITY, FITNESS FOR A PARTICULAR PURPOSE, NON INFRINGEMENT AND DATA
 * ACCURACY. THE AUTHOR AND CAC TEST LAB/DMDC DO NOT REPRESENT OR WARR-
 * ANT THAT THE OPERATION OF THIS SOFTWARE WILL BE UNINTERRUPTED OR ER-
 * ROR-FREE, OR THAT ANY DEFECTS WILL BE CORRECTED.
 */
import java.io.*;
import java.util.zip.*;
import java.security.cert.*;
import java.security.*;
import javax.smartcardio.*;

public class ApduUtilities{

    public static final byte[] PIV_EP_APP_AID = {(byte)0xA0, (byte)0x00,
(byte)0x00, (byte)0x03, (byte)0x08, (byte)0x00, (byte)0x00, (byte)0x10,
(byte)0x00, (byte)0x01, (byte)0x00};
    public static final byte[] PIV_EP_APP_CHUID_OID = {(byte)0x5F,
(byte)0xC1, (byte)0x02};

    /*
     * Convert byte array to equivalent hexadecimal string.
     */
    public static String binToHexString(byte[] buf, int length) throws
UnsupportedEncodingException{
        char [] chr = new char[(length*2)];
        String str;
        int h, l;
        char c;

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

    /*
     * convert byte array to equivalent hexadecimal string
     */
    public static String binToHexString(byte[] buf) throws
UnsupportedEncodingException{
        return binToHexString(buf,buf.length);
    }
}
```

Output from program execution:


C:\WINDOWS\system32\cmd.exe

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

CHUID:
5382078c3019d4f810da01156c109090458360da040c2bc0842182201093e2341030303030303030
010702a082073c30820738020103310b3009060520e03021a0500300a06086086480165030601a0
66310b3009060355040613025553311830160603550 40a130f552e532e20476f7665726e6d656e74
040313184 46f44204f4d20496e7465726d656469617465204341 2d32301e170d3039303630323134
3118301606035504 0a130f552e532e20476f7665726e6d656e74310c300a060355040b1303444f44
0403130e50532e4453532e4e482e5430303430820122300d06092a864886f70d0101010500038201
61c02b5c6dd28c347a43f0403f708bd8746c7ca8eefb54e3db1737c901f23f475827e8aabcfd1262
5496c64c129c871192a313d39b3868299392a3727e1831f85cf5edadfe575d887671b706ae79a953
04aadc2a59c1bed68549ce8548c917381249d9fed639d2214ba5c5659b46a9b7b489646cecb446c3
eadcc9235775df0203010001a382015e3082015a301f0603551d2304183016801 4751ab733de5bac
a1eee22a15ef45b17e30130603551d25040c300a06086086480165030607300e0603551d0f0101ff
1d1f044730453043a041a03f863d687474703a2f2f63726c2e6e69742e646973612e6d696c2f6765
30818a06082b06010505070101047e307c304a06082b06010505073002863e687474703a2f2f6372
6e7465726d6564696174652532304341 2d32302e06082b06010505073001862268747 4703a2f2f6f
0d0101050500038201010098c760181120883 4d7322b3693b268a77afec39bddcb88b0604b4bb44a
49ee55fe3d1e080228811f7e3aec74966193ee22b8784b75de42d99e2b464a86c9e352a894fab4d6
d3a87f5f39bd52b46c7dd69e2ccc078e2b8d98d9758de74d05830d353bfe864c4601a773313d548e
6781701d13c1af21c3b1da0abfa59040cb9a49f7d01ec5a2b01a68f75c2e9af697ddf9d04c16245a
0355040a130f552e532e20476f7665726e6d656e74310c300a060355040b1303446f44310c300a06
6961746520434 12d32020127300906052b0e03021a0500a081d03011060367 2a00310a0608608648
9a1ba2398206543078060860864801650306053 16c306a310b3009060355040613025553311830 16
44310c300a060355040b1303504b49310c300a060355040b13034f534431173015060355040313 0e
3031313231343135 5a300d06092a864886f70d0101010500048201008559cf77a6c68cfc2b975683
29b10352b33c5f3f1258d9dde6865f72b23cc0e644d22e1df5c970f83ef86e463e2601a3fb69020b
d5ed9f86b8e43632b9d87d8c64a07df31c684b9458f1ed674dc5f3f7b615018de0549120effc4868
ca4167f74bfb44bef7a0319713877635196 15ba22c6c25af8d288f481d11e299768b195de97699d7
press <Enter> to quit
```
```

Appendix F 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 one right index finger data. See INCITS 378 table 6 and SP 800-73-1 Appendix A for minutiae record descriptions.

		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
		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
		000C0A50	CBEFF product id
		0000	Capture equipment compliance and id
		0168	width in pixels
		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.
	Minutiae	8103005F3600	minutiae data
		810A00762D00	minutiae data
		….	….minutiae data
		….	….minutiae data
		0000	extended data (x00000 = none)
	F …	02	Position (right index finger).

		00	view number and impression
		FE	finger quality
		27	Number of minutiae.
	Minutiae	408F00A98900	minutiae data
		…	minutiae data
		…	minutiae data
		810700DE2F00	minutiae data
		80FB01078500	minutiae data
		…	minutiae data
		0000	extended data (x00000 = none)
CSB		55555555555555555555 55…..	CBEFF SIGNATURE BLOCK (CSB)

Appendix G 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 (# 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 H 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:

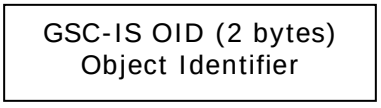
A0 00 00 01 16	DoD PIV Transitional GSC-IS 2.1 data model, also PIV data model as specified by Table 1 in Section 1.7 of SP 800-73-1
A0 00 00 00 79	DoD – CAC data model. The CCC follows the GSC-IS 2.1 (and PIV) data model
A0 00 00 03 08	NIST – PIV End-Point data model

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.

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).



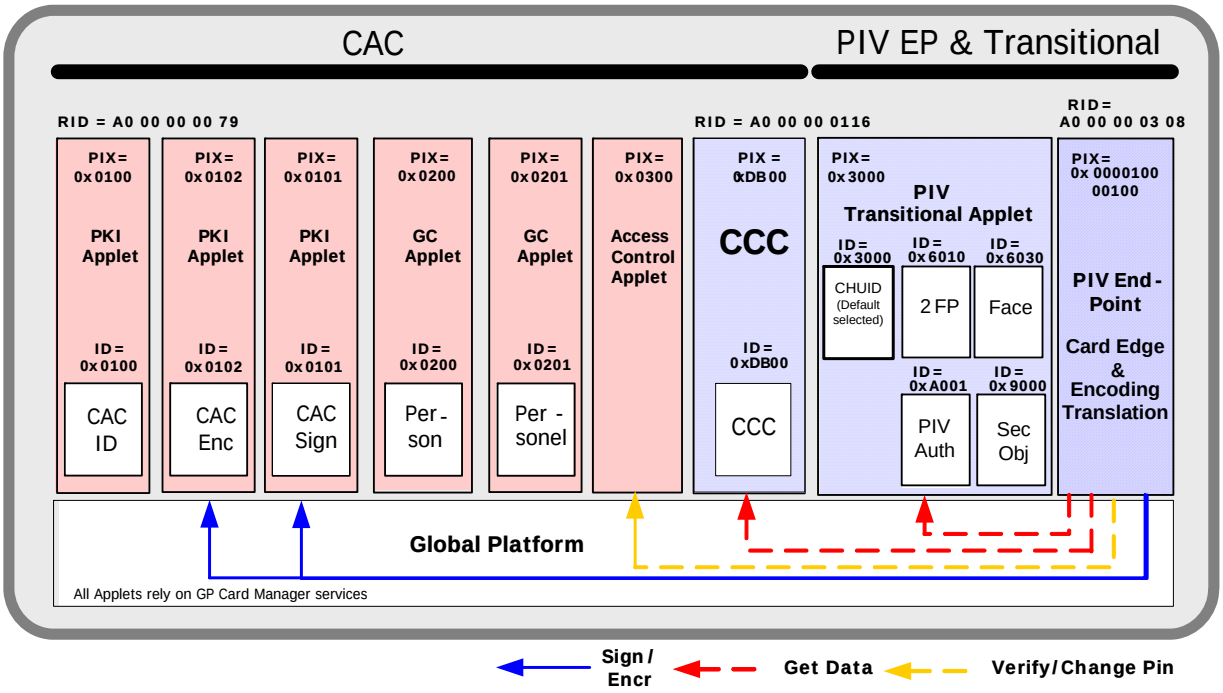
Appendix I CAC PIV End-Point 2.6.2 Applet Suite

This appendix contains information about the previous CAC PIV End-Point PIV End-Point data model and data mapping, referred to as the 2.6.2 Applet Suite. The CAC 2.6.2 applet suite and variations are applicable to the following CAC configurations:

- Oberthur ID One V5.2 (contact only)
- Gemalto Access 64KV2 (contact only)
- Oberthur ID One V5.2 Dual CAC PIV Transitional
- Gemalto GCX4 72K DI CAC PIV Transitional
- Oberthur ID One V5.2 Dual CAC PIV Endpoint
- Gemalto GCX4 72K DI CAC PIV Endpoint"

Data Model

Prior to the 2.6.2B applet, the following data model was used. At the time of writing, this data model is more prevalent than the consolidated data model accompanying the 2.6.2B applet. However, we expect to see a migration towards the consolidated data model (Sec. 4) in the near future.



Security Object Mapping of Data Groups to PIV Containers

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

DataGroup Number	container ID	dataGroup HashValue	References to SP 800-73-1	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 F)	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

Appendix J Private Sign/Decrypt APDU for RSA 2048

From GSC-IS 2.1 specifications.

5.3.6.1 Private Sign/Decrypt APDU

This command is used to perform an RSA signature or data decryption.

Command Message

CLA	0x80
INS	0x42
P1	0x00
P2	0x00
Lc	Data Field length
Data Field	Data to sign or decrypt
Le	Expected length of the signature/decryption

Data field sent in the command message

The data field contains the data to be signed using the selected RSA key pair.

The data must be already padded before the message is sent.

Response Message

Data field returned in the response message

The data field in the response message contains the data signed or decrypted. The client application is responsible for any data padding.

Processing state returned in the response message

See Table 5-11b.

Private Sign / Decrypt APDU for 2048 bit

Extend the command to support 2048-bit RSA:

P1 indicates whether more blocks containing the data to be signed, are to follow:

B_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Meaning
0	X	X	X	X	X	X	X	No more block to follow
1	X	X	X	X	X	X	X	More blocks to follow

The above extension is more GP aligned rather than ISO compliant (ISO uses CLA byte for command chaining).

Sample APDU:

80 42 80 00 80 + first block of 128 bytes
-> 9000
80 42 00 00 80 + first block of 128 bytes
-> 6100

00 C0 00 00 + bytes data to read
-> FF Bytes + 61 01

00 C0 00 00 (GET Response command until you read 9000)

Appendix K CAC Utilization and Variation Matrix

Below is snapshot of the CAC Utilization and Variation Matrix v1.81 July 28, 2009.

As of: 28 July 2009

Common Access Card (CAC) Platform—Utilization and Variation Matrix

Attributes	Obsolete Platforms			Current Platforms			
Card Manufacturer	Oberthur Card Systems (OCS) -Original Pilot CAC	Schlumberger (Axalto)	Oberthur Card System (OCS)	Axalto	Gemplus	Axalto	Oberthur Card Systems (contactless pilot)
Product Name	Galactic v1 32K	Cyberflex 32k v2 card with Softmask 7 Version 2	CosmopolIC v4 32K	Axalto Cyberflex Access 64k v1 soft mask 4 version 1	GemXpresso (GXP) PRO 64 K	Axalto Cyberflex Access 64k v1 soft mask 4 version 2	ID-One Cosmo v5.2D 64K
Laser Engraving on back of card	OCS Gal 2.1	Schlumberger Access 32K V2	Oberthur CosmopolIC v4	Axalto Access 64KV2	Gemplus GXP3 64V2N	Axalto Access 64KV2	Oberthur C.S. Cosmo64 V5.2D
Chip Size (EEPROM)	32K	32K	32K	64K	64K	64K	64K
Answer to Reset (ATR)	3B 7D 11 0000 00 31 80 71 8E 64 86D60200 82 9000	3b6500009c02 020702	3B 7F 11 0000 00 31 C0 53 CAC401 64 52D90400 82 9000	3B7512000029 05010401	3B6B00008065B 0830104748300 9000	3B75120000290 5010401	3B DB 96 00 80 1F 03 00 31 C0 64 77E30300 82 9000 C1
CAC Applet Package	V1 Applets	V1 Applets	V1 Applets	V2.3	V2.3 and V2.3.0c Applets	V2.3.0c Applets	V2.6.1 Applets
Other features*	Contact only RSA EE 1024	Contact only RSA EE 1024	Contact only RSA EE 1024	Contact Only RSA EE 1024	Contact only RSA EE 1024	Contact only RSA EE 1024	Dual interface RSA EE 1024
Operational Intro Date	Fall 2000	N/A	N/A	April 2005	N/A	N/A	May 2006
Operational Sunset Date	Spring 2004	End of Spring 2005	End of Spring 2005	Q1 CY 2007	Q1 CY 2007	Q1 CY 2007	Q2 CY 2007
Expiration of last card issued	End of Spring 2008	End of Spring 2008	End of Spring 2008	Q1 CY 2010	Q1 CY 2010	Q1 CY 2010	Q2 CY 2010

Page 1 of 3

As of: 28 July 2009

Card Manufacturer	Oberthur Card Systems	Gemalto	Oberthur Card Systems (PIV Transitional)	Gemalto (PIV Transitional)	Oberthur Card Systems (PIV Endpoint) with PIV endpoint applet and PIV auth cert	Gemalto (PIV Endpoint) with PIV endpoint applet and PIV auth cert	Gemalto (PIV Endpoint)
Product Name	ID-One Cosmo v5.2 72K	Cyberflex Access v2c 64K	ID-One Cosmo v5.2D 72K	Gemalto GemCombiXpresso R4 dual interface	ID-One Cosmo v5.2D 72K	Gemalto GemCombiXpresso R4 dual interface	Gemalto TOP DL GX4 144K
Laser Engraving on back of card	Oberthur ID One V5.2	Gemalto Access 64KV2	Oberthur ID One V5.2 Dual	Gemalto GCX4 72K DI	Oberthur ID One V5.2 Dual	Gemalto GCX4 72K DI	Gemalto TOP DL GX4 144K
Chip Size (ROM/EEPROM)	72K	64K	72K	72K	72K	72K	144K
Answer to Reset (ATR)	3B DB 96 00 80 1F 03 00 31 C0 64 77E30300 82 9000 C1	3B959540FF AE01030000	3B DB 96 00 80 1F 03 00 31 C0 64 77E30300 82 9000 C1	3B 7D 96 00 00 80 31 80 65 B0 83 11 13 AC 83 00 90 00	3B DB 96 00 80 1F 03 00 31 C0 64 77E30300 82 9000 C1	3B 7D 96 00 00 80 31 80 65 B0 83 11 13 AC 83 00 90 00	3B 7D 96 00 00 80 31 80 65 B0 83 11 17 D6 83 00 90 00 (Updated March 2009)
CAC Applet Package	V2.6.1 Applets	V2.6.1 Applets	V2.6.2 Applets	V2.6.2 Applets	V2.6.2 Applets	V2.6.2 Applets	V2.6.2b Applets
Other features*	Contact only RSA EE 1024	Contact only RSA EE 1024	Dual interface RSA EE 1024	Dual interface RSA EE 1024	Dual interface RSA EE 1024	Dual interface RSA EE 1024	Dual interface RSA EE 2048
Operational Intro Date	January 2007	February 2007	October 2006/ March 2007	June 2008	June 2008	June 2008	July/August 2009
Operational Sunset Date	Ongoing	Ongoing	Ongoing	Ongoing	Ongoing	Ongoing	TBD
Expiration of last card issued	Ongoing	Ongoing	Ongoing	Ongoing	Ongoing	Ongoing	TBD

Page 2 of 3

As of: 28 July 2009

	Emerging Platforms
Card Manufacturer	Oberthur Card System (PIV Endpoint)
Product Name	Oberthur ID-One Cosmo 128 v5.5 for DoD CAC
Laser Engraving on back of card	Oberthur ID One 128 v5.5 Dual
Chip Size (ROM/EEPROM)	128K
Answer to Reset (ATR)	3B DB 96 00 80 1F 03 00 31 C0 64 B0 F3 10 00 07 9000 80
CAC Applet Package	V2.6.2b Applets
Other features*	Dual interface RSA EE 2048
Operational Intro Date	Target (Winter 2009)
Operational Sunset Date	TBD
Expiration of last card issued	TBD

*includes the key strength of RSA end-entity (EE) certificates with SHA-1 that are available for specific platforms/configurations.

Page 3 of 3

Index