

DMDC

Card Technologies & Identity Solutions Division (CTIS)



**DoD Implementation Guide for Transitional
PIV II SP 800-73 v1**

Version 1.01

March 2006

Revision History Page

Issue Date	Document Version	Modification By	Change Description
Aug 15, 2005	Version 0.08	Robert van Spyk	Draft
Aug 22, 2005	Version 0.09	Chris Khan	Security Object
Oct 1, 2005	Version 0.19	Dave Wasmuth	Review,
Oct 15, 2005	Version 0.29	Robert van Spyk	Draft
Nov 12, 2005	Version 0.35	Dave Wasmuth	Testing and compliance
Nov 21, 2005	Version 0.40	Robert van Spyk	PACS integration
Jan 6, 2006	Version 0.50	Robert van Spyk	Integration of IAB concerns and other comments
Feb 9, 2006	Version 0.51	Marlon A. Guarino	Integration of DMDC comments
Feb 23, 2006	Version 0.51	Lynne Prince	Changes
Mar 5, 2006	Version 0.54	Robert van Spyk	Cert and Biometric
Mar 22,2006	Version 0.57	Marlon A. Guarino	Data model discovery,Cert and Biometric
Mar 24,2006	Version 1.0	Robert van Spyk	Update for 800-73 v 1

TABLE OF CONTENTS

1 INTRODUCTION 1

 1.1 BACKGROUND 1

 1.2 PURPOSE 1

 1.3 AUDIENCE 1

 1.4 THE DoD CAC ENVIRONMENT 1

 1.5 ASSERTIONS 2

2 INTEROPERABILITY USE CASES 2

 2.1 INTEROPERABILITY USE CASE SCENARIOS 2

 2.1.1 *Use Case 1: DoD requires physical access to non-DoD federal facility*..... 2

 2.1.2 *Use Case 2: Non-DoD Agency requires physical access to DoD facility*..... 3

 2.1.3 *Use Case 3: DoD requires logical access to non-DoD federal systems*..... 3

 2.1.4 *Use Case 4: Non-DoD Agency requires logical access to DoD system*..... 3

 2.2 USE CASE OPERATIONAL CONSTRAINTS 3

 2.3 PIV AND CAC COMPONENTS 3

3 DATA MODEL DISCOVERY 5

 3.1 DATA MODEL DISCOVERY COMBINATIONS 5

 3.2 DATA MODEL DISCOVERY FOR CONTACT CARD 5

 3.3 DATA MODEL DISCOVERY FOR CONTACTLESS 6

4 TRANSITIONAL PIV DATA MODEL 7

 4.1 VERSION NUMBER 8

 4.2 TRANSITIONAL DATA ENCODING 8

 4.3 ADDRESSING OF DATA OBJECTS 10

5 DOD TRANSITIONAL PIV DATA ELEMENTS 11

 5.1 CCC 12

 5.1.1 *CCC Requirements* 12

 5.1.2 *CCC Syntax*..... 12

 5.1.3 *CCC CAC v2 and PIV Buffer Content* 13

 5.1.4 *CCC Usage* 15

 5.1.5 *CHUID*..... 16

 5.1.5.1 *CHUID Usage*..... 16

 5.1.5.2 *FASC-N*..... 18

 5.1.5.3 *Global Unique Identifier (GUID)* 21

 5.1.5.4 *Expiration Date*..... 21

 5.1.5.5 *Authentication Key Map*..... 21

 5.1.5.6 *Error Detection Code*..... 21

 5.1.5.7 *Issuer Asymmetric Signature*..... 22

 5.1.6 *Fingerprint Buffer 0x6010*..... 23

 5.1.7 *Security Object* 25

 5.1.7.1 *Security Object Specification* 26

 5.1.7.2 *Mapping of Data Groups to PIV Containers*..... 27

 5.1.7.3 *LDS Security Object*..... 27

 5.1.7.4 *Authenticate PIV Container with Security Object* 28

 5.1.8 *Facial Image Buffer*..... 28

 5.1.9 *X.509 Certificates and keys for PIV Authentication and CAC PKI Signature*..... 28

 5.1.10 *PIV Authentication Key* 29

6 PIV TRANSITIONAL BSI 30

7 TRANSITIONAL PIV CARD EDGE 30

 7.1 CONTACTLESS INTEROPERABILITY 31

8 BACK-END SYSTEM TRANSACTIONS 31

8.1 VALIDATION TRANSACTIONS	32
8.2 TRANSACTION USE CASES.....	33
9 CONFORMANCE TESTING.....	34
9.1 CAC AND CAC WITH TRANSITIONAL PIV IMPLEMENTATION CONFORMANCE TESTING.....	34

LIST OF APPENDICES

APPENDIX A	ACRONYMS AND REFERENCES.....	35
APPENDIX B	ICAO PROFILE LDS SECURITY OBJECT	37
APPENDIX C	MESSAGE DIGEST HASH ALGORITHMS.....	38
APPENDIX D	COMPARISON OF SIMPLE TLV AND BERT TLV.....	39
APPENDIX E	STANDARD BIOMETRIC HEADER (CBEFF_HEADER).....	40

LIST OF FIGURES

FIGURE 1	USE CASES DIAGRAM	4
FIGURE 2	CONTACTLESS CARD DISCOVERY FLOW.....	6
FIGURE 3	CONTACT CARD DISCOVERY FLOW	7
FIGURE 4	T-BUFFER FORMAT.....	9
FIGURE 5	V-BUFFER FORMAT	9
FIGURE 6	SIMPLE TLV	9
FIGURE 7	GSC-IS 2.1 CCC ENCODING	13
FIGURE 8	FASC-N FORMAT	19
FIGURE 9	FASC-N EXAMPLE	19
FIGURE 10.	SECURITY OBJECT STRUCTURE.....	26
FIGURE 11	SECURITY OBJECT TRANSACTION	28
FIGURE 12	PIV ISSUER VERIFICATION FOR LOGICAL ACCESS	33

LIST OF TABLES

TABLE 1	CARD DATA MODEL DISCOVERY	5
TABLE 2	CAC APPLICATION CARD URL VALUES	16
TABLE 3	PIV APPLICATION CARD URL VALUES	16
TABLE 4	CHUID CONTAINER	16
TABLE 5	FASC-N DATA ELEMENTS.....	18
TABLE 6	AGENCY CODE	19
TABLE 7	SECURITY OBJECT CONTAINER	26
TABLE 8	SECURITY OBJECT CONTAINER ELEMENTS	26
TABLE 9.	MAPPING OF DG TO CONTAINER ID.....	27
TABLE 10.	LDS SECURITY OBJECT BINDINGS.....	27
TABLE 11.	PIV, CAC KEY, AND CERTIFICATE ACCESS RULES.....	29
TABLE 12	FILE CONTROL INFORMATION	31
TABLE 12	HASH ALGORITHM REQUIREMENTS FOR THE 800-73 SECURITY OBJECT.....	38
TABLE 13	HASH ALGORITHM OBJECT IDENTIFIERS FOR THE 800-73 SECURITY OBJECT	38

DoD Implementation Guide for transitional PIV II SP 800-73

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 Encryption [PKI] programs are being aligned to meet this additional set of requirements.

1.2 Purpose

This Guide specifies technical details for implementing interagency PIV I and PIV II National Institute of Standards and Technology [NIST] Special Publication [SP] 800-73v1 requirements in the DoD CAC environment. It documents how the DoD CAC and middleware are implemented with PIV. This Guide includes mandatory and PIV optional but DoD mandatory PIV capabilities.

Usage, policy, and CACv2 specifics are out of scope for this document but are discussed in other documents. The information in this document is subject to change although to the degree possible, the DoD standardizes the issuance and post-issuance process so that all vendors can reference a common set of criteria for PIV.

1.3 Audience

This guidance applies to the SP 800-73 and those who provide, acquire, test or develop applications, middleware or applets for the DoD Smart Card program.

1.4 The DoD CAC Environment

The PIV solution is implemented on the DoD CAC, but is separate and distinct from the DoD multifunction CAC. It will evolve at its own pace but in the same environment.

The purpose and function of the CAC is much broader than the narrowly focused interoperability function of the PIV. 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 (1) provide for logical access to computer systems, (2) provide personnel identification, (3) enable physical access to buildings, and (4) PKI for signing, encryption, and non-repudiation. The CAC is the standard identification card for active duty military personnel, Selected Reservists, DoD civilian employees, and eligible contractor personnel.

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

The Identity Protection and Management Senior Coordinating Group (IPMSCG) is responsible for integrating cross-functional requirements and determining summary-level chip storage allocations, to include those for Component-specific use of the CAC. The group coordinates claims against Joint-Service CAC resources and platform functionality with the managers whose missions are supported by the CAC.

1.5 Assertions

- Scope of this document is determined by use cases
- Scope is the transitional interfaces and data model as described in sections 1 and 2 of SP 800-73
- The PIV and CAC applications have clearly defined dependencies.
- CAC is the primary application for DoD
- DoD will add PIV mandatory data model elements to the CAC
- This guide focuses on Java Card implementation
- Optional SP 800-73 elements may be mandated for internal use by DoD
- Interoperability within DoD is out of scope of PIV
- No harm to existing middleware and card
- Middleware will have the ability to communicate with CAC and PIV
- CAC Credentials will be the primary Credentials

This document represents the delta between the CACv1/CACv2 and the FIPS 201. The CAC platform baseline requirements, host application, issuance process, are covered elsewhere; they are only mentioned here when required as context.

2 INTEROPERABILITY USE CASES

Interoperability refers to the interoperable use of identity credentials among federal agencies and departments. The relying party authenticates the issuing agency credentials presented to it. This authentication may require validation against issuing agency information. Upon success, the relying party then registers the individual's credentials in its local system along with the appropriate authorizations. Physical or logical access applies the individual's credentials to the local system, possibly again with validation against the issuing agency if warranted by policy.

Interoperability applies to both the card and the systems that authorize physical or logical access. Backend transactions among agencies play an important role in interoperability of identity credentials. They support the validation of the credentials against issuing departments during registration and usage. Equally important, they may provide information about usage back to the issuing party. This includes not only revocation, loss or misuse, but also changes in security conditions and other events that must be recorded by the issuing party for maintaining identity credential integrity.

The use cases below define the requirements for DoD PIV compliance as determined by use cases.

2.1 Interoperability Use Case Scenarios

This section presents a set of inter-agency personal authentication scenarios that demonstrate how the DoD will interoperate with other federal agencies.

2.1.1 Use Case 1: DoD requires physical access to non-DoD federal facility

In this scenario, a DoD cardholder presents a DoD CAC SP 800-73 PIV compliant card. The card is a CAC v2 dual interface card with a PIV transitional, typically used in contactless mode with PIN and/or biometric used for card-holder authentication, depending on the non-DoD agency security policy. Local physical access credentials may be issued but these are not stored on the card. Only the CHUID is read over the contactless interface for physical access.

The non-DoD facility's application or middleware must support the PIV transitional card edge for the DoD card to interoperate successfully. The relying agency must have access to DoD

back-end systems to validate the DoD CHUID and obtain any additional data such as issuer signature CRL and employee status.

2.1.2 Use Case 2: Non-DoD Agency requires physical access to DoD facility

In this scenario the transitional dual interface or end-point PIV card is presented to the DoD. Registration requires all or part of the PIV data model to surface at the contact interface. Usage will typically be in contactless mode with PIN and/or biometric used for card-holder authentication.

The DoD supports the GSC-IS BSI and card edge for the PIV transitional card, thus leveraging existing middleware. The PIV host application is written to the GSC-IS BSI. Non - DoD agencies must support the transitional card edge and data model for interoperability. Vendors may support both the transitional and end-point for physical access.

2.1.3 Use Case 3: DoD requires logical access to non-DoD federal systems.

A DoD CAC cardholder presents a DoD-issued SP 800-73 PIV transitional card to gain logical access to another agency's resources, using a contact reader. The card is a CAC v2 card with a PIV application. The card will be used strictly in contact mode for both registration and use.

The non-DoD facility's application and middleware must support the PIV transitional card edge for the DoD card to interoperate successfully. DoD back-end systems must allow validation by the relying party of DoD credentials. None-DOD agency policy will set the security policy in terms of registration and use.

2.1.4 Use Case 4: Non-DoD Agency requires logical access to DoD system.

The cardholder presents a transitional or end-point PIV card. The transitional card is supported. If the card is a PIV end-point, some form of cross-credentialing like Contractor Verification System (CVS) and Defense Cross Credentialing Identification System (DCCIS) may be used for registration. Issuer back-end systems must allow validation by the DoD of credentials presented.

2.2 Use Case Operational Constraints

The above use cases are theoretical. In practice, usage will be much more constrained. Logical access requires considerable IT involvement to provide the appropriate authorization privileges. Prior to access, the card and card holder must be authenticated and registered. This requires the card holder and identity information to exist in a back-end system or to be added to it. Authorization is likely granted in a manner of a new employee. Registering and access sites may be able to process both transitional and end-point credentials depending on vendor implementation.

Physical access via contactless is the common scenario. Card readers are addressed directly via the card edge and because of the very limited application may accommodate both transitional PIV cards and PIV end-point cards.

2.3 PIV and CAC Components

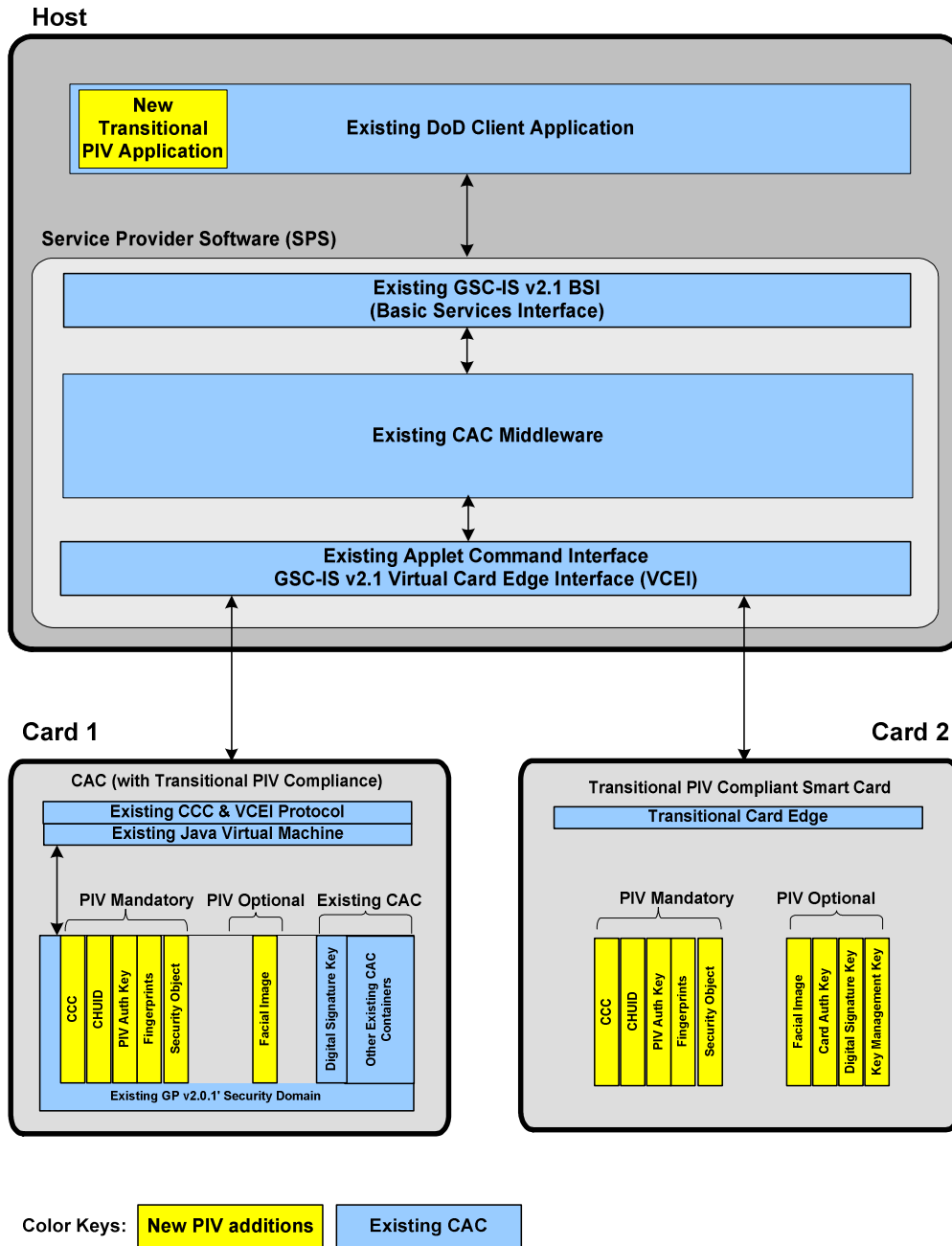
Figure 1 below illustrates the components in a PIV smart card solution. The large box on top represents a DoD computer hosting CAC or PIV applications. The two cards beneath it represent different transitional PIV cards that might be presented to the DoD host.

Card 1 in Figure 1 below illustrates the CAC with PIV transitional compliant solution. This solution leverages the existing GSC-IS 2.1 [GSC-IS] BSI and card edge to serve existing CAC and PIV applications. A PIV host application will use the PIV for physical or logical access.

Card 2 illustrates the non-CAC transitional card. A PIV host application will use the PIV for physical or logical access, communicating via the existing GSC-IS 2.1 [GSC-IS] interfaces. In both cases the host application will be written to the BSI.

Figure 1 Use Cases Diagram

One Approach to supporting CAC and Transition Smart Cards.



3 DATA MODEL DISCOVERY

The data model corresponds to the scope and version of data objects.

CAC and PIV functionality may be presented at the contact interface requiring data model discovery. This section presents an informative example of the discovery process.

SP 800-73 v1 notes that conditions based on access mode (contact vs contactless) take precedence over access rules in Table 2.

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)
CACv2	No CCC implemented.	N/A
Contactless Pilot	CCC (GSC-IS RID)	0x02
CAC with transitional PIV	CCC (GSC-IS RID)	0x10
Transitional PIV compliant card	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 a discovery procedure for identifying the data models on a contact card for logical access. Note that the order of discovery steps is informative.

Select CCC.

IF CCC not found THEN

Select ACA, get Properties (*using GSC-IS 2.1 Specs*).

IF Select ACA AND get Properties SUCCESS THEN

RETURN CAC V2.

ELSE IF Select CAC V1 SUCCESS THEN

RETURN CAC V1

ELSE IF CCC AID = Transitional AID THEN

IF data model id tag 0xF5 = 0x02 THEN

RETURN Contactless Pilot

ELSE IF data model id tag 0xF5 = 0x10 THEN

RETURN PIV Transitional

ELSE IF CCC AID = PIV End-Point AID AND data Model ID = 0x10 THEN

RETURN PIV End-Point.

RETURN Unknown data model

3.3 Data Model Discovery for Contactless

As shown below four contactless card types will be deployed in the field. 1, 2 and 3 offer the same contactless card interface. Card 4 utilizes a different contactless interface.

1. Contactless Pilot CAC with Contactless Pilot card profile with GSC-IS interfaces.
2. CAC with PIV transitional card profile with GSC-IS interfaces.
3. PIV transitional card. from another agency with GSC-IS interfaces.
4. PIV end-point card from another agency with PIV data model and end-point interfaces.

Discovery of the PIV transitional vs. end-point is based on the selection of the default container. The default container for contactless is the CHUID. The PIV CHUID is read with the READ BINARY card edge command. If that command is not supported, PIV end-point can be assumed. READ BINARY response message is in Simple TLV format.

Overall Discovery flow diagrams for contact and contactless cards:

Figure 2 Contactless Card Discovery Flow

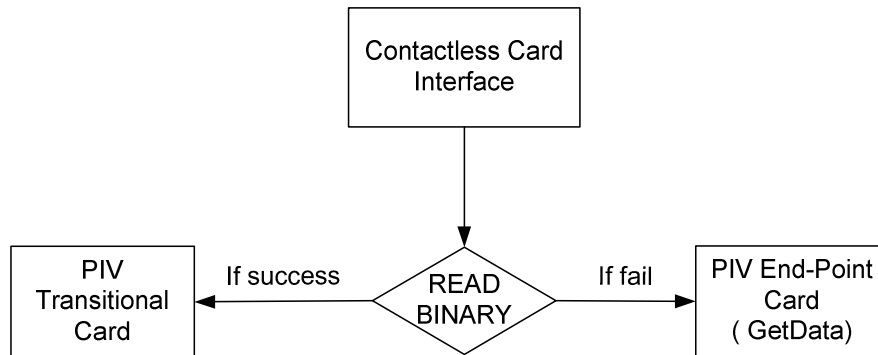
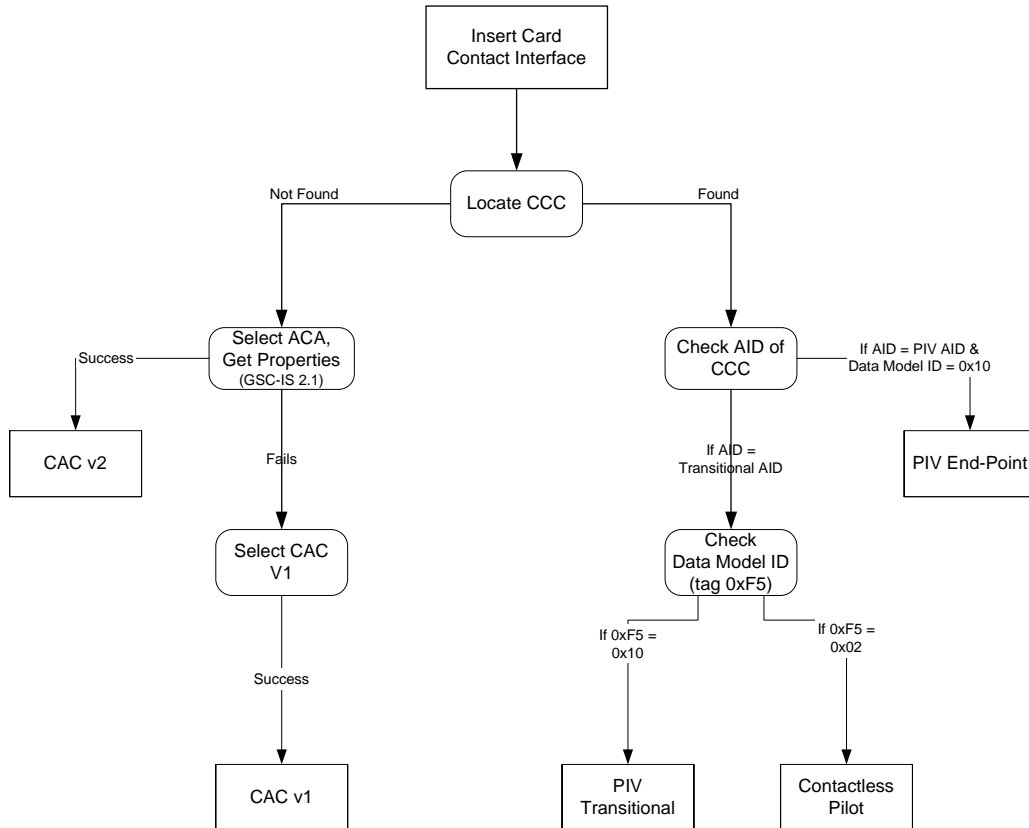


Figure 3 Contact Card Discovery Flow



4 TRANSITIONAL PIV DATA MODEL

The PIV data model content and access control are common between the PIV transitional and PIV end-point definitions. It includes six mandatory containers and five optional containers. The DoD will implement those listed below, including two of the optional containers. This section describes the DoD PIV data model. This data model contains all PIV mandatory objects and elements, as well as optional SP 800-73 specs that are DoD mandatory.

DoD implements the following PIV containers:

- CCC
- CHUID
- Security Object
- Card Holder Fingerprint Buffer (containing primary and secondary fingerprints).
- Facial Image Buffer
- PIV Auth Key

The DoD will use the CAC PKI Signing Key as the PIV Auth Key so no separate PIV key will be generated.

The remaining optional PIV containers are not implemented:

- Digital Signature Key (Existing CAC PKI Identity key)
- Key Management Key (Existing CAC PKI Encryption key)
- Card Authentication Key (optional)

The DoD transitional data model containers and access rules are as follows:

Table 2 SP 800-73 v1 DoD Data Model Containers

RID 'A0 00 00 01 16'	Container ID	Access Rule	Contact/Contactless	M/O
Card Capability Container	0xDB00	Read Always	Contact	M
CHUID Buffer	0x3000	Read Always	Both	M
PIV Auth Certificate Buffer	0x0101	Read Always	Contact	M
Fingerprint Buffer	0x6010	PIN	Contact	M
Facial Image Buffer	0x6030	PIN	Contact	O, DoD Mandatory
Security Object Buffer	0x9000	Read Always	Contact	M

4.1 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** – CACv1 and CACv2
- **0x02** – the Contactless Pilot which is CACv2 plus the CHUID and CCC
- **0x10** – the DoD CAC with PIV Transitional or End-Point

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

4.2 Transitional Data Encoding

Section 8 of the GSC-IS 2.1 (NISTIR 6887) specifies that data will be encoded using Simple TLV format. The CAC data model adheres to this standard. Since 1) neither FIPS 201 nor part 2 of SP 800-73 mentions encoding and 2) section 2.3.2, *Data Format and Structure*, of SP 800-73 refers to NISTIR 6887, Chapter 8, PIV data model shall be implemented using Simple TLV. This section provides a brief description of Simple TLV, as defined in Chapter 8 of NISTIR 6887.

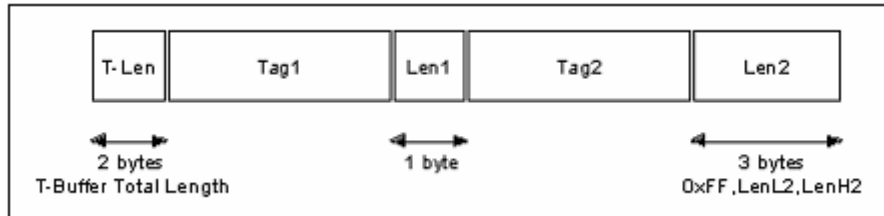
PIV applets are implemented as Generic Containers (GC) according to GSC-IS 2.1[GSC-IS]. Tags, globally unique values data element identifiers, are 1 byte encoding only number from 1 to 254. Tags 0x00 and 0xFF are invalid. Scope of tags is at the container level. Buffer length is encoded in either 1 or 3 bytes and is the length of the Value component. If the data length is greater than 255, then the Length is encoded in 3 bytes, the first byte being the head byte with value FF. Therefore, the Value can be up to 64K in length.

The GC applet contains one or multiple sets of two buffers, one per GSC-IS container. The buffer lengths are defined at instantiation. One buffer contains the Tags and the length of the values, and the second buffer contains the values. This permits different access rights. The buffers are implemented such that the T-buffer length = (T-buffer length-2 bytes), and the V- buffer length = (V-buffer length – 2 bytes).

In the case of the CAC, the terminal or host-side application manages the content of the buffers. The terminal can write to the buffer if it has permissions. To create new space to store data, a new instance of the GC applet is created by the issuer.

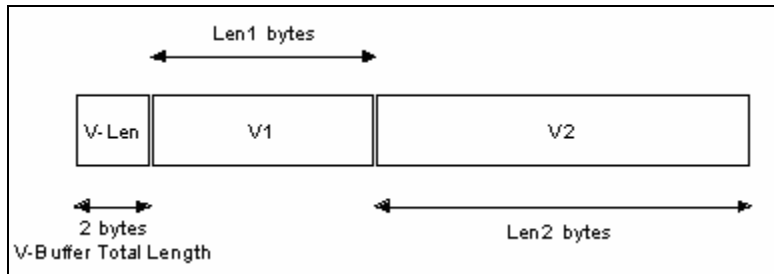
CAC tags are 1 byte in length. The T-Buffer and V buffer shall be constructed as follows according to the TLV format:

Figure 4 T-Buffer Format



The CAC V-Buffer shall be constructed as follows according to the TLV format:

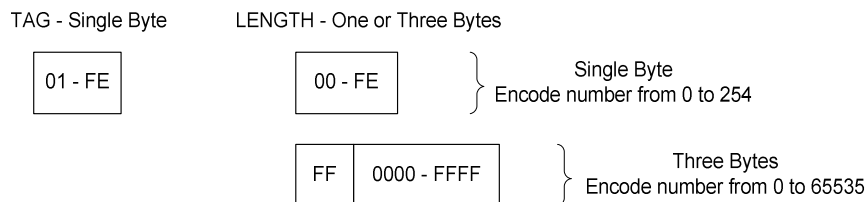
Figure 5 V-Buffer Format



PIV transitional is encoded with SIMPLE TLV. PIV end-point specifies the use of Basic Encoding Rules (BER-TLV). Both are defined in ISO 7816-4¹. For informational purposes, SIMPLE TLV is shown in the Figure 6 diagram below.

Figure 6 Simple TLV

SIMPLE TLV



ISO/IEC 8825-1:2002, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

¹ For informational purposes, the difference between SIMPLE and BER TLV see Appendix D

4.3 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

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

AID – Application Identifier

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
- 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 and PIV objects are referenced with a 7 byte **Universal AID** as follows:

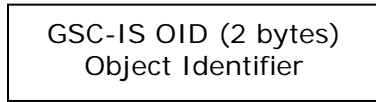
RID (5 bytes) Resource Identifier	GSC-IS OID (2 bytes) Object Identifier
--------------------------------------	---

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, 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.

RID (5 bytes) Resource Identifier	PIX (0-11 bytes) Proprietary Identifier Extension
--------------------------------------	--

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



5 DoD TRANSITIONAL PIV DATA ELEMENTS

The PIV data model containers that are implemented by the DoD on the PIV-enabled CAC are further defined in the following sections.

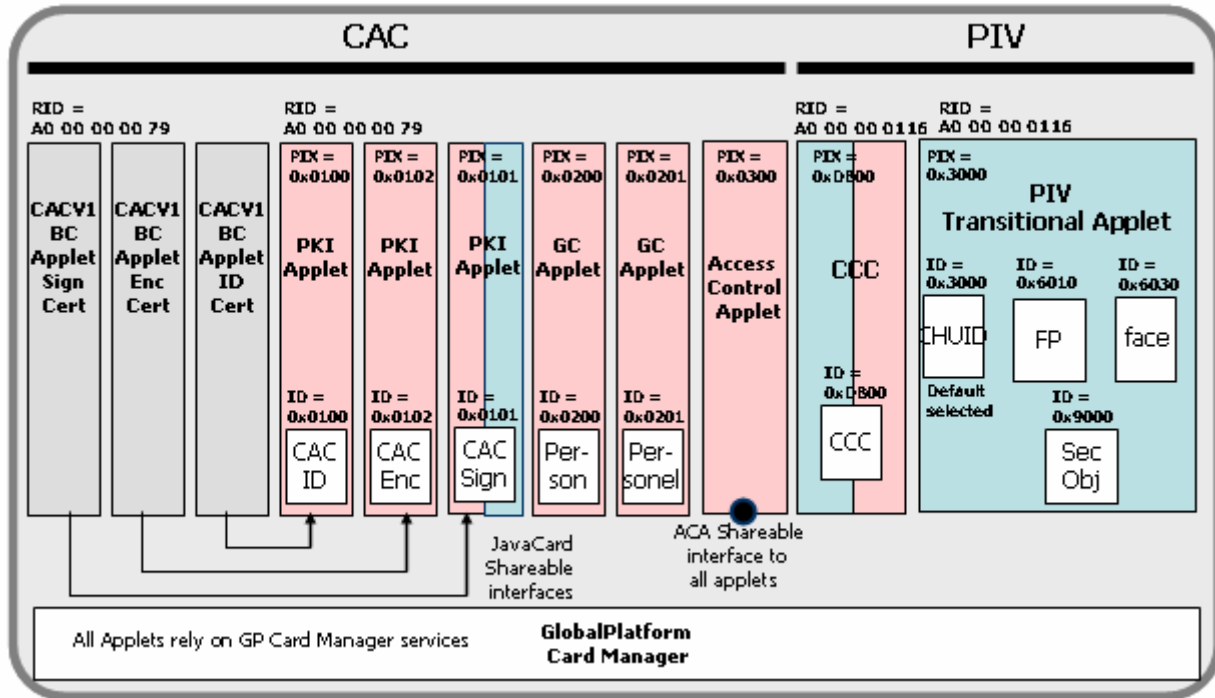
DoD implements the following PIV containers:

- CCC as specified in SP 800-73
- CHUID
- Security Object
- Card Holder Fingerprint Buffer (containing primary and secondary fingerprints).
- Facial Image Buffer
- PIV Auth key

The remaining PIV containers are:

- Digital Signature Key (Existing CAC PKI Identity key)
- Key Management Key (Existing CAC PKI Encryption key)
- Card Authentication Key (optional)

Figure 7 CAC and PIV profile



800-73 permits default applets. In the above profile the default selected applet for both contact and contactless is the CHUID. Its PIX and ID are the same and hence SELECT OBJECT is not required and only READ BINARY is needed.

5.1 CCC

5.1.1 CCC Requirements

The CCC must be presented at the contact interface. It is in compliance with the SP 800-73 transitional specification, and supports minimum capabilities for lookup on data model and application information

To allow fast and light discovery of card capabilities, there is a single CCC on the card when both PIV and CAC Applications are present. The data model number is 0x10. The CCC is the default applet in contact card.

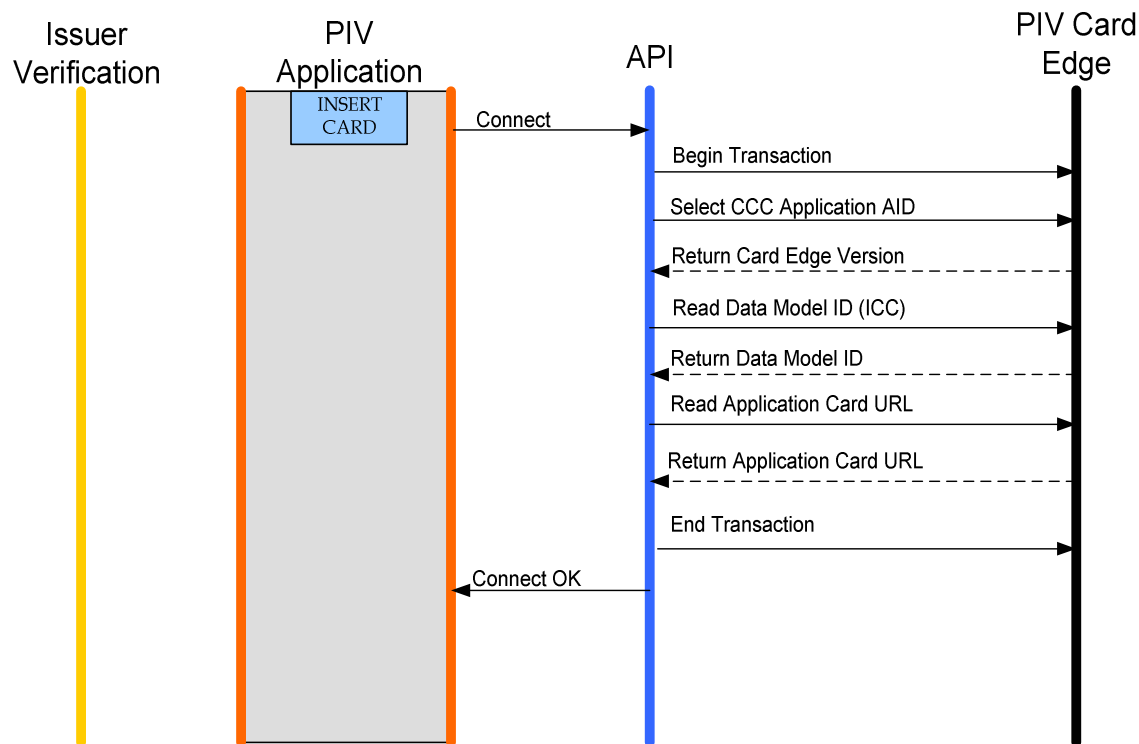
5.1.2 CCC Syntax

The DoD CCC supports minimum capabilities for lookup on data model and application information. The CCC definition in Appendix A of SP 800-73 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. 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.

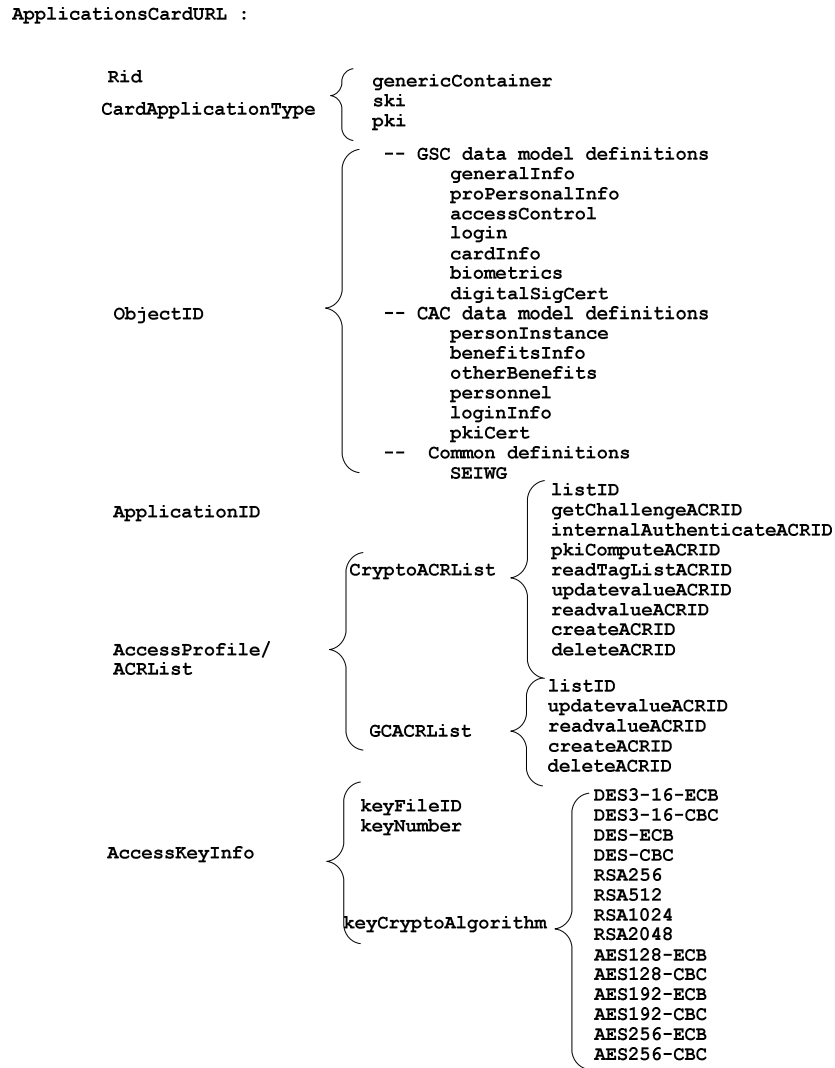
Retrieval of CCC Information

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 and GSC-IS 2.1. The components and the values allowed in a card URL are depicted in the figure below.

Figure 7 GSC-IS 2.1 CCC encoding



5.1.3 CCC CAC v2 and PIV Buffer Content

CCC AID is fixed as **A000000116 DB00** (ref. GSCIS v2.1 §6.2)

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)
	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 JavaCard; '01' for FileSystem 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) 01 for GC; 04 for PKI;
ObjectID (2B) e.g. 0x02FE for pki certificate...
AppID (2B) same as ObjectID for CAC objects
AccProfile not used for JavaCard, 00
PinID (1B) not used for JavaCard, 00
AccKeyInfo (4B) not used for JavaCard, 00000000

See Table 1 and 2 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 00000000000000000000 (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)

5.1.4 CCC Usage

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.

The tables below list the application URL entries used by the DoD. Current GSC-IS and PIV CCC URL entries are 16 bytes each. SP 800-73 suggests a maximum of 128 bytes for the variable ApplicationsCardURL data element, leaving room for only 8 total entries, which is too small to contain all of the entries needed by the DoD. That limit is interpreted as a guideline, not a hard limit.

Table 2 CAC Application Card URL values

RID	App ID	Object ID	Description	Application Card URL
A000000079	0200	0200	Person	F3 10 A000000079 01 0200 0200 00 00 00000000
A000000079	0201	0201	Personnel	F3 10 A000000079 01 0201 0201 00 00 00000000
A000000079	0100	0100	PKI-ID	F3 10 A000000079 04 0100 0100 00 00 00000000
A000000079	0102	0102	PKI-Encryption	F3 10 A000000079 04 0102 0102 00 00 00000000
A000000079	0101	0101	PKI-Signing (PIV AUTH)	F3 10 A000000079 04 0101 0101 00 00 00000000

Table 3 PIV Application Card URL values

RID	App ID	Object ID	Application	Application Card URL
A000000116	3000	3000	Card Holder Unique Identifier (CHUID)	F3 10 A000000116 01 3000 3000 00 00 00000000
A000000116	3000	6010	Card Holder Fingerprint Buffer	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

5.1.5 CHUID

The DoD follows the PIV requirement.

5.1.5.1 CHUID Usage

The CHUID is free read from both the contactless and contact interface.

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

Table 4 CHUID Container

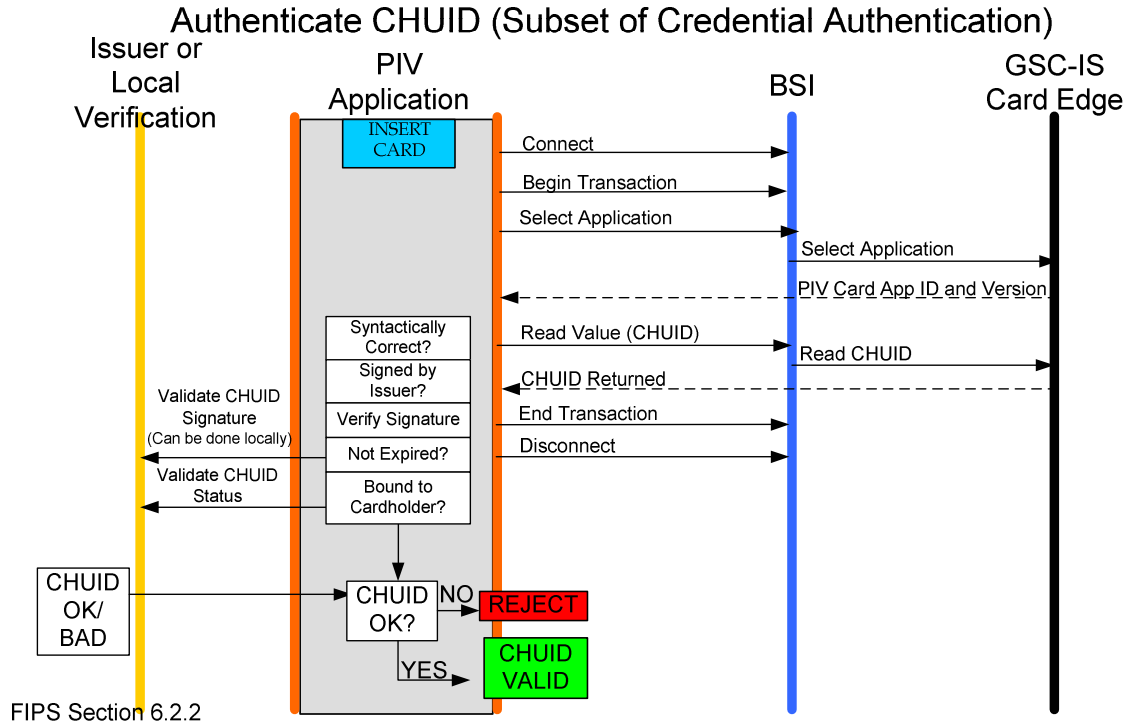
Card Holder Unique Identifier		0x3000	Always Read	
Data Element (TLV)	Tag	Type	Max Bytes	M/O

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 following pertain to the CHUID:

- The CHUID includes an element, the Federal Agency Smart Credential Number (FASC-N), which uniquely identifies each card.
- The PIV CHUID shall be accessible from both the contact and contactless interfaces of the PIV Card without card activation.
- The FASC-N shall not be modified post-issuance.
- In addition to the mandatory FASC-N that identifies a PIV Card, the CHUID shall include an expiration date.
- In machine readable format, the expiration date data element shall specify when the card expires. The expiration date format and encoding rules are as specified and formatted SP800-73. This date is the same as that on the printed surface and the Printed Information Buffer.
- This standard requires inclusion of the Asymmetric Signature field in the CHUID container
- The Asymmetric Signature data element of the PIV CHUID shall be encoded as a Cryptographic Message Syntax (CMS) external digital signature, as defined in RFC 3852 [RFC3852].
- The digital signature shall be computed over the concatenated contents of the CHUID, excluding the Asymmetric Signature field (FIPS 201 4.2.2) and the Authentication Key Map, if present (See Appendix).
- SP 800-73 states “dual interface VM cards shall have the CHUID Object available for selection in the default selected applet allowing them to honor a Select Object/EF CHUID issued immediately after the card answer to reset. “
- Algorithm and key size requirements for the asymmetric signature are detailed in [SP800-78].

The CHUID may be recovered and authenticated as shown in the flow diagram below.



5.1.5.2 FASC-N

The Federal Agency Smart Credential Number (FASC-N) shall be consistent with the TIG SCEPACS Option for "System Code || Credential Number" to establish a credential number space of (Number of RAPIDS stations x 99,999,999) credentials (SP 800-073).

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.

The FASC-N is comprised of the following elements:

Table 5 FASC-N Data Elements

Field Name	Abbreviation	Length (BCD digits)
Agency Code	AC	4
System Code	SC	4
Credential Number	CN	6
Credential Series	CS	1
Individual Credential Issue	ICI	1
Person Identifier	PI	10
Organizational Category	OC	1
Organization Identifier	OI	4
Person/Organization Association Category	POA	1
Start Sentinel	SS	1

Field Separator	FS	1
End Sentinel	ES	1
Longitudinal Redundancy Character	LCR	1

The format of the FASC-N is as follows:

Figure 8 FASC-N Format

SS	AC	FS	SC	FS	CN	FS	CS	FS	ICI	FS	PI	OC	OI	POA	ES	LRC
----	----	----	----	----	----	----	----	----	-----	----	----	----	----	-----	----	-----

An example of a DoD FASC-N would be the following:

Figure 9 FASC-N Example

```

11010 10011 11100 00001 00001 10110 00001 00001 00001 10000 10110
SS 9 7 0 0 FS 0 0 0 1 FS

10000 00001 00001 00001 11100 00001 10110 00001 10110 10000 10110
1 0 0 0 7 0 FS 0 FS 1 FS

10000 00001 00001 00001 00001 00001 00001 00001 00001 00001 10000
1 0 0 0 0 0 0 0 0 0 1

01000 10000 00001 00001 00100 11111 00100
2 1 0 0 4 ES 4
    
```

5.1.5.2.1 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. The following table outlines the possible Agency Code values based on the sites Affiliation. For all cards issued by DoD these values are as defined in Table 6. The DoD has no alphanumeric AC.

Table 6 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

5.1.5.2.2 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. A new mapping will be developed that will provide a 4 digit CHUID site identifier corresponding to DMDC's established 6 digit site identifiers. A similar mapping will have to be developed if the DBIDS site identifier is not a 4 digit number. To ensure uniqueness across the DoD enterprise RAPIDS and DBIDS must not share System Codes.

5.1.5.2.3 Credential Number,

The Credential Number, Credential Series and Individual Credential Issue fields will be used to define the tokens Credential Number. The combination of an Agency Code, System Code and Credential Number is a fully qualified number that is uniquely assigned to a single individual.

For RAPIDS and DBIDS a one up numbering scheme per site will be used to ensure uniqueness. With the three fields, when used in conjunction with the Credential Series there are 99,999,999 credential numbers per site.

5.1.5.2.4 Token, Alternate Credential Number, Number Option

Guidance allows for an alternate solution for the Credential Number. For the DoD, each site will have a stored counter that will start at 100,000,001. With each card issued the stored counter will be read, incremented and used to populate the Credential Number, Credential Series and Individual Credential Issue fields. The 6 rightmost digits of the stored counter will occupy the Credential Number. The seventh digit will be the Credential Series and the eighth digit will occupy the Individual Credential Issue.

5.1.5.2.5 Agency Code, System Code, Token Credential Number Implementation Series

The Agency Code, System Code, and Token Credential Number table definition will be as follows:

TABLE CHUID_CNSTRCT	
SITE_ID	NUMBER(6) NOT NULL
SITE_ORG_CD	CHAR(2) NOT NULL
AGNCY_CD	NUMBER(4) NOT NULL
CHUID_SITE_ID	NUMBER(4) NOT NULL
CRDNTL_NMBR	NUMBER(10) NOT NULL

The DoD shall use the Credential number to indicate the multiple of one million cards that have been issued at a particular site. This value shall start at 0 and be switched to 1 once 99,999,999 cards have been issued at a particular site.

5.1.5.2.6 Individual Credential Issue

The Individual Credential Issue (ICI) value is initially encoded as a "0".

5.1.5.2.7 Person Identifier

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

5.1.5.2.8 Organizational Category

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

- 1 – Federal Government Agency
- 2 – State Government Agency
- 3 – Commercial Enterprise
- 4 – Foreign Government

5 – Locally Assigned

The DoD will use 1 or 5 in this field.

5.1.5.2.9 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. The RAPIDS Agency/Sub-agency table will be modified to provide a mapping from the DMDC Agency Codes to the SP 800-87 Organization codes. DBIDS will use a similar table.

5.1.5.2.10 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:

- 1 – Employee
- 2 – Civil
- 3 – Executive Staff
- 4 – Uniformed Service
- 5 – Contractor
- 6 – Organization Affiliate
- 7 – Organization Beneficiary

5.1.5.3 Global Unique Identifier (GUID)

The GUID enables migration away from the FASC-N into a robust numbering scheme for all issued credentials. It is currently defined as an issuer assigned IPv6 address. The DoD will populate this field with 16 zeros.

5.1.5.4 Expiration Date

Expiration date is found in the CHUID, printed on the card, and contained in the Printed Information Buffer. These are the same dates and refer to the expiration date of the card. The format is YYYYMMDD.

5.1.5.5 Authentication Key Map

The Authentication Key Map is an optional field that enables an external application to obtain a key reference for implementing a symmetric key challenge/response protocol using the Card Authentication Key. The Authentication Key Map, if present, is not signed, as it may be modified by the local PACS system. The DoD does not plan to implement this capability.

5.1.5.6 Error Detection Code

The Error Detection Code (LRC) will be determined over the CHUID.

5.1.5.7 Issuer Asymmetric Signature

The DoD CHUID contains the issuer asymmetric digital signature. The DoD will retrieve a digital certificate from a Defense Information Systems Agency [DISA] approved Certificate Authority.

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 processing rules in RFC3852 apply:

- 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

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

- A binary string representing the plain-text concatenated from all of the following mandatory and optional elements (if present)“

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.

Value		Comments
SignedData		
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. The Public Key is included but the issuer certificate is I the CHUID
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	
version	m	Version must be 1 because of mandated sid choice. (See RFC3852 Section 5.3 for rules regarding this field).
sid	m	
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 encapsulatedContent and 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		
MessageDigest	m	The chuidSecurityObject is hash over security object.
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	The result of the signature generation process.
unsignedAttrs	o	Issuers may wish to use this field, but it is not recommended and receivers may choose to ignore them.

5.1.6 Fingerprint Buffer 0x6010

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]; any other biometric data agencies elect to place on PIV Cards such as the Facial Image.

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

Table 7 Simple CBEFF Structure

Data Element	References
CBEFF_HEADER	INCITS 398 5.2.1, SP800-76 6
CBEFF_BIOMETRIC_RECORD	SP800-76 section 6, INCITS 398 5.2.2
Patron format PIV	SP800-76 section 6
Finger Minutia Record	SP800-76 section 3.3 , INCITS 398 5.2.2, SP800-76 Table 3 (fields 12-36 only)
CBEFF_SIGNATURE_BLOCK	FIPS 201 4.4.2, INCITS 398 5.2.3

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

No extended data.

No proprietary data.

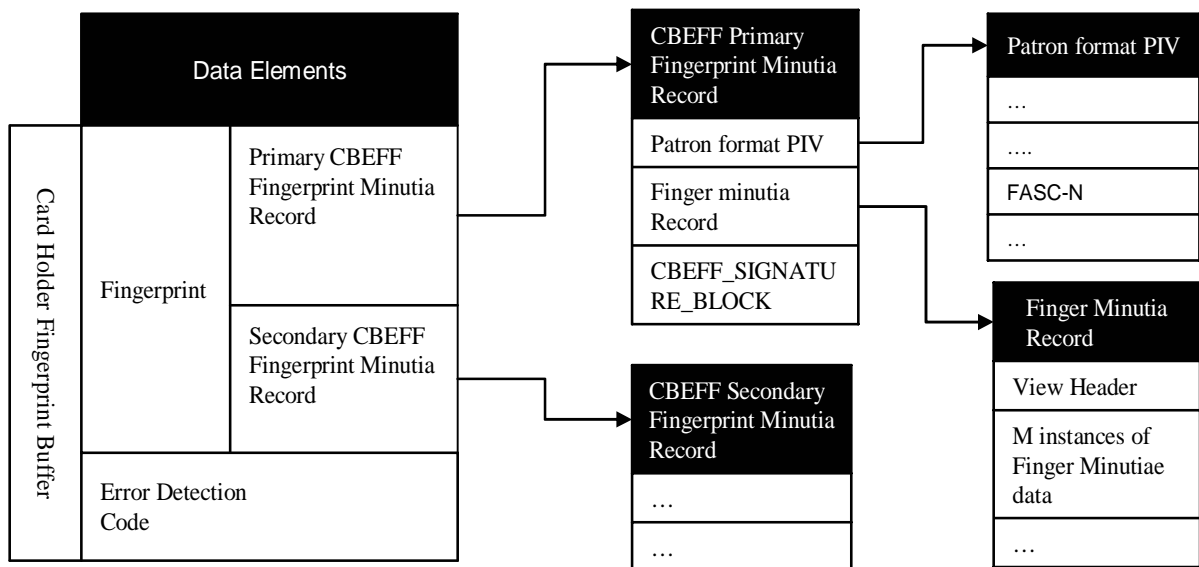
Each finger minutia record has only one view.

Restriction of minutia type (bifurcation, ridge ending).

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

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

Table 8 Card Holder Fingerprints Buffer Structure.



Further details on fields of this structure are specified on SP800-76 table 3. Fields contained in this structure are from table 3, starting from rows 12 – 36. Rows 1 – 11 give normative practices for PIV.

The fingerprint buffer contains two fingerprints. The CBEFF wrapper does not explicitly indicate the presence of one or two fingerprints in the buffer. At the time of this document, SP800-73v1 did not specify how to access the secondary fingerprint minutiae.

5.1.7 Security Object

The principal goal of the security object is to reduce the overhead of cryptography in the data model. 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. Note that the PIV Security Object is different from the ICAO Security Object in that it contains not only the object hashes but also the DG mapping with PIV containers. The Security Object contains hashes of the Facial Image, which is specifically called out in SP 800-73.

The facial image buffer will not change over the life cycle of the card.

If a container is not present on the card, the container IDs mapped to their respective Data Groups will be null.

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. The issuer's certificate is not included with the Security Object, since it is already part of the CHUID. Normative signature reference is RFC 3852.

When the Facial Image Buffer (0x6030) is present on the card, a hash of its contents and the algorithm used will be present in the Security Object.

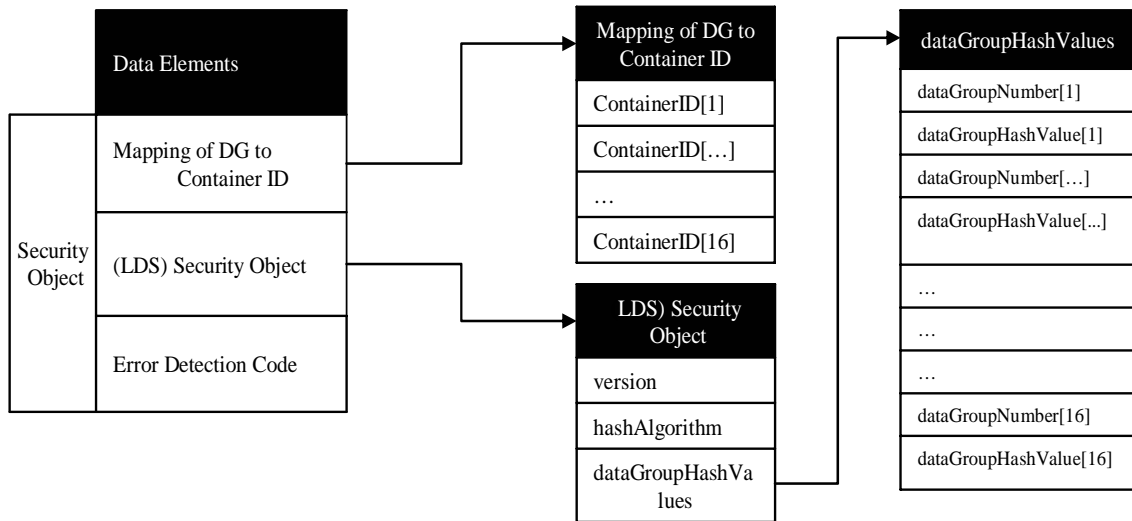
The current contents of the Facial Image Buffer (0x6030) can be read from the card, hashed using the algorithm identified in the Security Object, and compared to the hash value present in the Security Object.

If the two hash values are identical and the signature is verified, the integrity of the current contents of the desired buffer is valid and can be confidently used to identify or authenticate that the card belongs to the card holder.

The CAC PKI Signature Key (PIV Auth Key) is signed by the CA. This signature may be validated. Hence the security object will be present but not utilized.

The figure below shows the structure layout of the Security Object.

Figure 10. Security Object Structure.



Note: In SP800-73, the data structure for “Mapping of DG to Container ID” is not defined. The mapping here is an array of 16 elements with the DG numbers implicit.

5.1.7.1 Security Object Specification

The SP 800-73 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 in Appendix B..The hashes are those in the signature of each signed object. The hashes are placed in the security object in the order in which data elements are presented in the PIV data model overview. The issuer’s certificate hash is not included in the Security Object element, since it is already part of the CHUID.

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

Table 7 Security Object Container

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

Table 8 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

5.1.7.2 Mapping of Data Groups to PIV Containers

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.

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

Table 9. Mapping of DG to Container ID

DataGroupNumber	Container ID	dataGroupHashValue	References to SP800-73	Comment
1	0x0000	MIT		
2	0x6030	MIT	Image for Visual Verification (Appendix A)	Similar to LDS Encode Face Datagroup [MRTD].
3	0x6010	MIT	Card Holder Fingerprints (Appendix A)	Similar to LDS Encode Finger Datagroup [MRTD].
4	0x0000	null		
5	0x0000	MIT		
6	0x0000	null		
7	0x0000	null		
8	0x0000	null		
9	0x0000	null		
10	0x0000	null		
11	0x0000	null		
12	0x0000	null		
13	0x0000	null		
14	0x0000	null		
15	0x0101	MIT	Authentication Public Key	Similar to Active Authentication Public Key Info. Datagroup [MRTD]
16	0x0000	null		

[MIT] Mandatory at time of instantiation.

5.1.7.3 LDS Security Object

Table 10. LDS Security Object Bindings.

(LDS) Security Object 0xBB (Variable Max Length 900 bytes)			
Data Element	Type	Value	Comment
version	Integer	0	TBD
hashAlgorithm	AlgorithmIdentifier	1.3.6.1.5.5.7.0.18	See note 1 below.
dataGroupHashValues	dataGroupHash	dataGroups [16]	16 Data groups according to [MRTD].

Note:

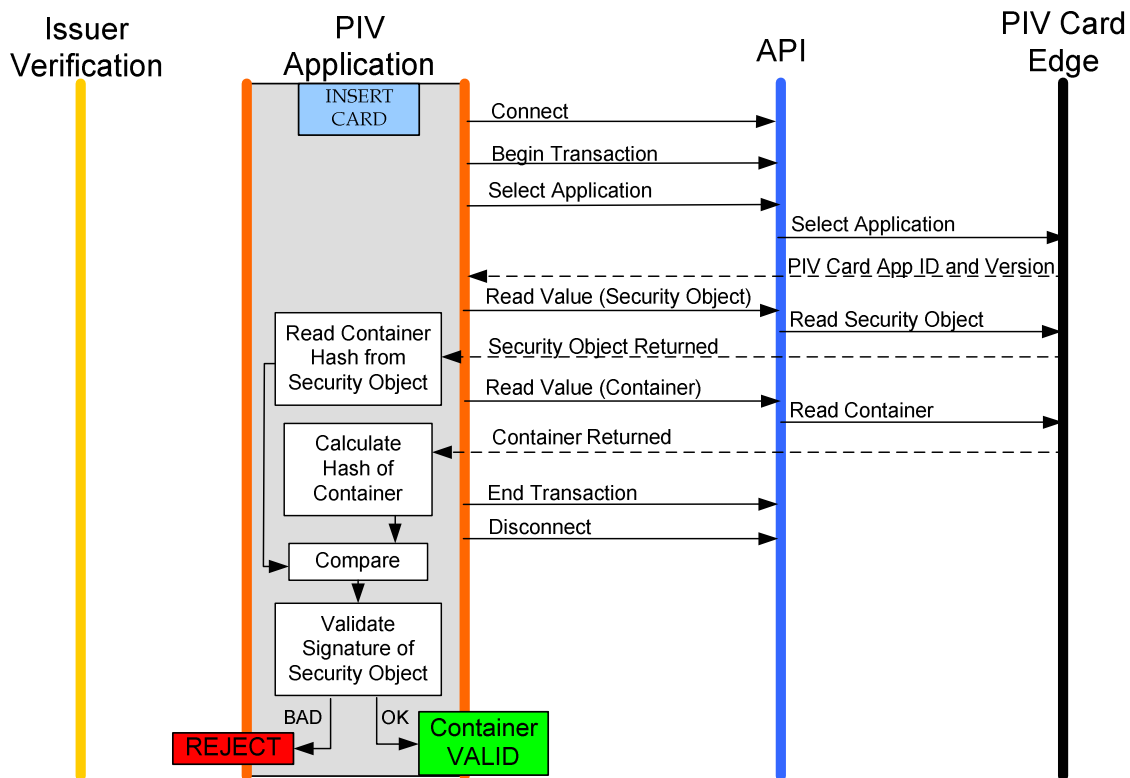
1. AlgorithmIdentifier comes from PKIX1Explicit88 { iso(1) identified-organization(3) dod(6)internet(1) security(5) mechanisms(5) pkix(7)mod(0) pkix1-explicit(18). [RFC 3280]

5.1.7.4 Authenticate PIV Container with Security Object

The diagram below shows how the security object may be used to authenticate a PIV container.

Figure 11 Security Object Transaction

Authenticate PIV Container with Security Object



5.1.8 Facial Image Buffer

The facial image stored in this buffer is not intended for electronic facial recognition systems. It is only intended to be used for human visual verification.

5.1.9 X.509 Certificates and keys for PIV Authentication and CAC PKI Signature

The table below exhibits the access control rules for PIV and CAC key usage. The right column lists the keys in use by the DoD today. The left column lists the optional and mandatory keys proposed by the PIV standard. The DoD PKI Signature Key will converge with the PIV Authentication Key and serve as such. PIV optional keys will not be implemented.

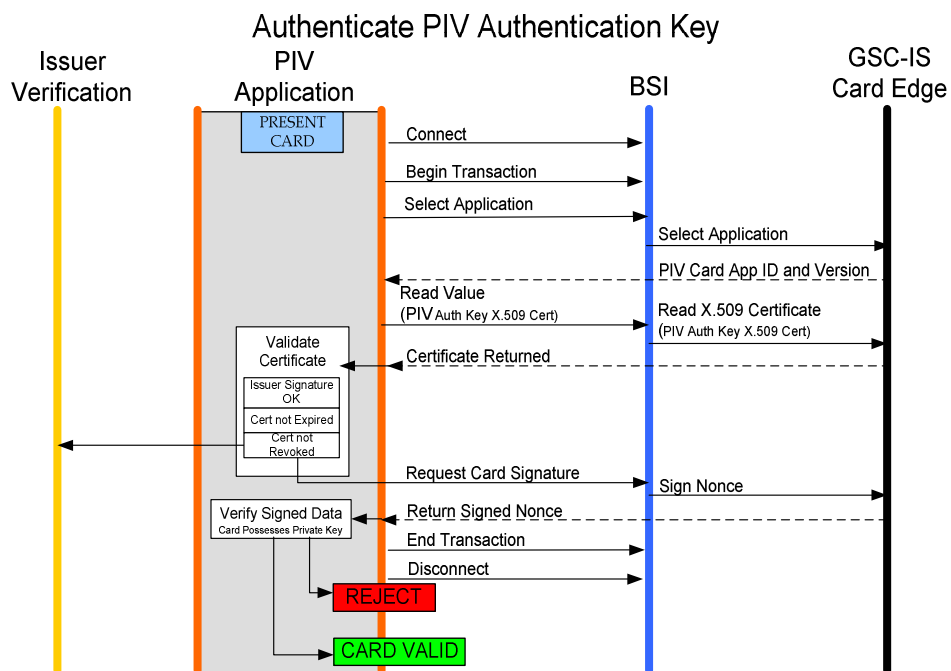
Table 11. PIV, CAC Key, and Certificate Access Rules.

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 CH using PIN. Identity key for logical access.	ALW/PIN	0x0101	M	PKI Signature Key	PKI Logical Login (Outlook) Digital Signature with non-repudiation, logical access, PIN. Outlook requires special extension.	ALW/PIN	0x0101	M
Digital Signature Key	Digital Signature for non-repudiation. Contact only	ALW/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	ALW/PIN	0x0102	O	PKI Encryption Key	Key Encipherment	ALW/PIN	0x0102	M

Note: The gray area in the table indicates keys which the DoD does not plan to implement since they already exist within the DoD PKI (except the PIV Auth Key).

5.1.10 PIV Authentication Key

This single key is used to authenticate the card holder for logical access scenarios. Within the DoD, the CAC Signature key and associated certificate is being used in Microsoft cryptographic logon process. The PIV Auth key may be authenticated as follows:



6 PIV TRANSITIONAL BSI

The PIV API is implemented as a subset of GSC-IS 2.1 BSI as specified in SP 800-73.

Following is a list of these functions that are required to implement the PIV application use cases defined in this document:

- gscBsiUtilAcquireContext()
- gscBsiUtilConnect()
- gscBsiUtilDisconnect()
- gscBsiUtilBeginTransaction()
- gscBsiUtilEndTransaction()
- gscBsiUtilGetVersion()
- gscBsiUtilGetCardStatus()
- gscBsiUtilGetExtendedErrorText()
- gscBsiUtilGetReaderList()
- gscBsiUtilReleaseContext()
- gscBsiGcReadTagList()
- gscBsiGcReadValue()
- gscBsiPkiCompute()

7 TRANSITIONAL PIV CARD EDGE

The PIV card edge is implemented as a subset of GSC-IS 2.1 card edge as specified in SP 800-73 Part I.

The subset of GSC-IS card edge commands implemented by the PIV card application are listed below. These commands apply to VM cards.

Commands for common interface:

- SELECT APPLETT / SELECT OBJECT
- GET RESPONSE

Card platform commands for common interface:

- READ BUFFER
- READ BINARY

Commands for authentication:

- VERIFY
- PRIVATE SIGN / ENCRYPT

In addition to the specifications in GSC-IS, SP 800-73 contains the following requirements:

- When an applet is selected, the response message contains the minimum File Control Information defined in ISO 7816-4 (FCI), as follows:

Table 12 File Control Information

Offset	Value	Description
00h	6Fh	FCI template tag
01h	4 + AID Length	Length of FCI template
02h	84h	Application name tag
03h	AID Length	Length of application name
04h	AID	Instance AID Value
4+ AID Length	A5h	Proprietary Data tag
5+ AID Length	00h	Length=00

- The PIN used in PIV Cards using the File Card Edge shall comply with the PIN format defined in Section 3.5.3 of SP 800-73, i.e., if the PIN length is less than 8 bytes, it shall be padded to 8 bytes with 'FF' bytes.

SP 800-73 allows use of the GSA AID (A0 00 00 01 16 DB 00) as the name of the CCC to find out if it is a file card.

Access Control is supported by PIV and utilized according to policy.

7.1 Contactless Interoperability

The CHUID is presented at the contact and the contactless interface.

SP 800-73 contactless cards provide a minimum interoperability mechanism for cardholder identification in both physical access control and contact card type environments. This cardholder identification mechanism should use the same card edge functions (APDUs) and Data Models as those defined for GSC-IS 2.1 contact cards, to ensure interoperability with middleware designed for GSC contact cards.

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 and DoD will support it in end-Point when applets, card edge, and middleware are available. In the meanwhile physical access device vendors may support both READ BINARY and GET DATA for interoperability.

800-73 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.

8 BACK-END SYSTEM TRANSACTIONS

Although the SP 800-73 does not explicitly state a requirement for back-end System validations for interoperability, the DoD asserts this in the only sure way to validate the legitimacy of the cardholder or the cardholder credential. Clearly, an agency would not want to grant access to an individual, either logically or physically, simply on the basis of a person presenting a token. The Back-end systems transactions are the insurance that the

cardholder's affiliation is still valid and that the card itself has not been revoked. This section illustrates how each of the use cases cited in section 2.1 could use a back-end transaction to strength the identity and authentication of a PIV implementation.

PIV prescribes CRL and OCSP as the mechanism for checking the validity of a credential during the authentication process. This assumes that all validity checking during usage for access only involves certificates, that all CAs are constantly available and responsive enough to provide CRL checking, and that CRLs are up to date. In addition, required data for on-system validation, such as CHUID and biometric elements, will need to be available. Without such validation mechanisms the card and credentials would be assumed to be good.

8.1 Validation Transactions

Use cases here apply only to issuer validation of contact cards. Contactless card validation is a subset. FIPS 201 and SP 800-73 assume the following usage capabilities:

Physical Access - For physical access the following events take place:

1. Verify CHUID
2. Verify card holder PIN/Biometric (Optional)

Logical Access - For logical access the following events take place:

1. Verify card holder PIN
2. Verify Digital Certificate/Digital Signing

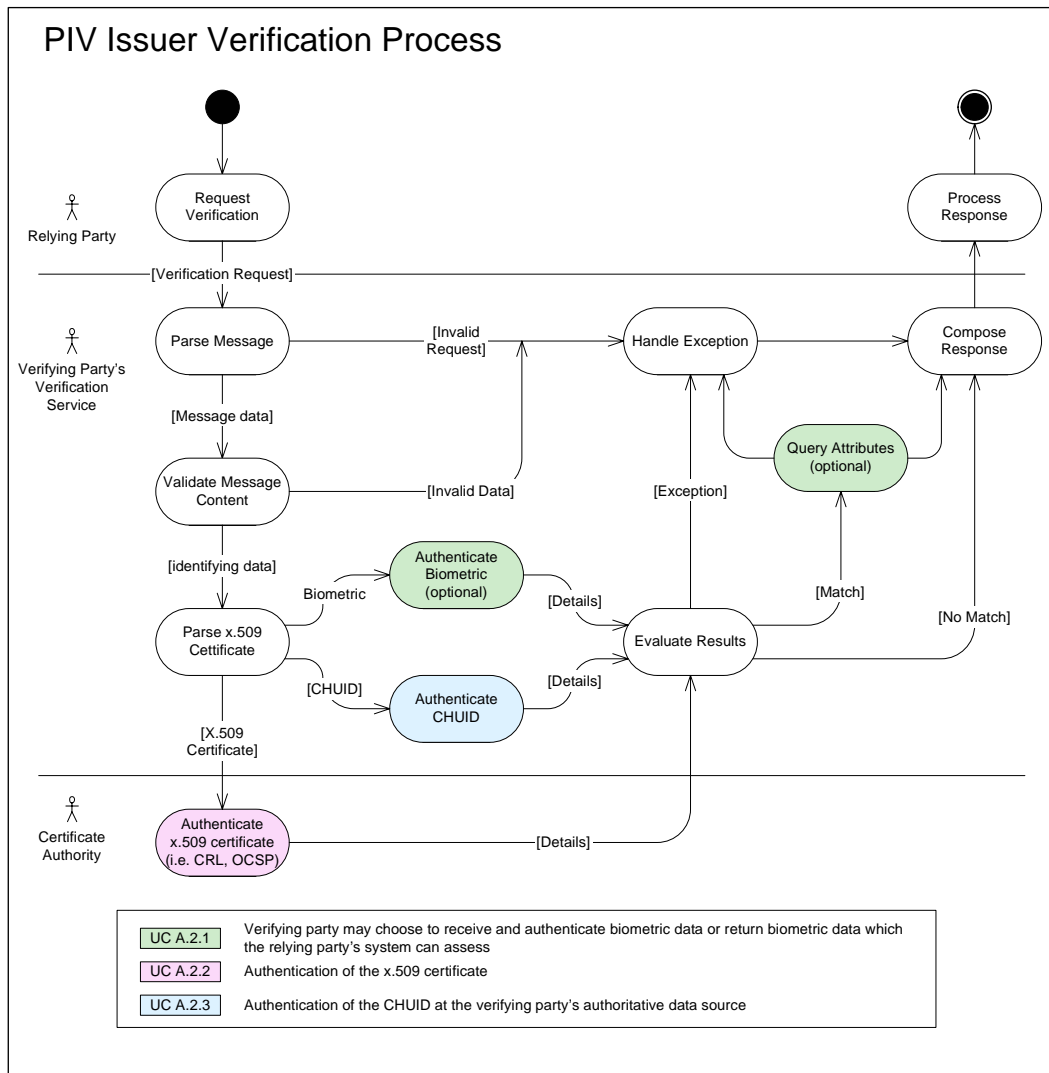
Each use case transaction may result in external (Issuer) verification.

Figure 11 represents the verification logic from the relying party request to the issuer response. Note that this is an illustration only and the logic as well as implementation may vary.

Internal issuer processes will be transparent. However, in generating responses the following decision logic is likely to be employed.

Each Relying party verification request and the validation response of good/bad, or data, will likely be transmitted using a standard protocol such as SAML.

Figure 12 PIV Issuer Verification for Logical Access



The figure above represents issuer capability to respond on-line to compound Security Assertion Markup Language (SAML) requests. This solution is preferred in that it guarantees freshness of primary sources (but not necessarily back-end sources), and uniform inter-department transmission protocol.

8.2 Transaction Use Cases

Examples of back-end transactions are as follows. There is ongoing work on architecture and specifications.

1. Request Issuer to validate by passing a CHUID or X.509 certificate and the Card ID from the card.
2. Request Issuer to validate by passing a CHUID or X.509 certificate and a fingerprint from the card.

3. Request Issuer to validate by passing a CHUID or X.509 certificate and receiving back a Photo of the Cardholder.
4. Request Issuer to validate PIV Digital Certificate by passing an X.509 certificate from the card.

9 CONFORMANCE TESTING

9.1 CAC and CAC with Transitional PIV implementation conformance Testing

The DoD recognizes that the current SP 800-73 has no provision for conformance testing of the PIV II implementation (Transitional state). Given this omission, the DoD has augmented its testing facility to include testing of the GSC-IS 2.1 card edge and BSI commands as well as the contactless interface.

Appendix A Acronyms and References

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
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
RID	Registered Identifier

[CBEFF] Common Biometric Exchange File Format

[FIPS 201] NIST *Federal Information Processing Standards Publication 201-1, Personal Identity Verification for Federal Employees and Contractors*, 2 September, 2005.

[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.

[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] NIST Special Publication 800-73, *Integrated Circuit Card for Personal Identity Verification*, NIST, February 2005.

[SP800-76] NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification*, NIST, February 2005.

[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.

[SP800-85] NIST Special Publication 800-85, *Draft NIST Special Publication 800-85, PIV Middleware and PIV Card Application Conformance Test Guidelines*, NIST, October 2005.

[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 B ICAO Profile LDS Security Object

```

LDSSecurityObject {iso(1) identified-organization(3) icao(ccc) mrtd(1)
security(1) ldsSecurityObject(1)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- Imports from RFC 3280 [PROFILE], Appendix A.1
AlgorithmIdentifier FROM
PKIX1Explicit88 { iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7)
mod(0) pkix1-explicit(18) }
-- Constants
ub-DataGroups INTEGER ::= 16
-- Object Identifiers
id-icao OBJECT IDENTIFIER ::= {1.3.ccc }
id-icao-mrtd OBJECT IDENTIFIER ::= {id-icao 1}
id-icao-mrtd-security OBJECT IDENTIFIER ::= {id-icao-mrtd 1}
id-icao-ldsSecurityObject OBJECT IDENTIFIER ::= {id-icao-mrtd-security 1}
-- LDS Security Object
LDSSecurityObjectVersion ::= INTEGER {V0(0)}
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
LDSSecurityObject ::= SEQUENCE {
    version LDSSecurityObjectVersion,
    hashAlgorithm DigestAlgorithmIdentifier,
    dataGroupHashValues SEQUENCE SIZE (2..ub-DataGroups) OF
DataGroupHash }
DataGroupHash ::= SEQUENCE {
dataGroupNumber DataGroupNumber,
dataGroupHashValue OCTET STRING }
DataGroupNumber ::= INTEGER {
dataGroup1 (1),
dataGroup2 (2),
dataGroup3 (3),
dataGroup4 (4),
dataGroup5 (5),
dataGroup6 (6),
dataGroup7 (7),
dataGroup8 (8),
dataGroup9 (9),
dataGroup10 (10),
dataGroup11 (11),
dataGroup12 (12),
dataGroup13 (13),
dataGroup14 (14),
dataGroup15 (15),
dataGroup16 (16)}
END

```

Notes:

The 'ccc' in id-icao defines the ICAO organization. The value of this field (defined by the Registration Authority for ISO 6523) to be published by ICAO.

The field dataGroupValue contains the calculated hash over the *complete* contents of the Data group EF, specified by dataGroupNumber.

Appendix C Message Digest Hash Algorithms

The following table identifies the hash algorithms specified in SP 800-78 as the algorithms that may be used for creating the message digests (hashes) of information on the card. The set of acceptable algorithms depends upon the expiration date of the CAC Card, since the hash algorithm must protect the data during the entire card lifetime.

Table 13 Hash Algorithm Requirements for the 800-73 Security Object

Card Expiration Date	Algorithm
Through 12/31/2010	SHA-1, SHA-224 or SHA-256
After 12/31/2010	SHA-224 or SHA-256

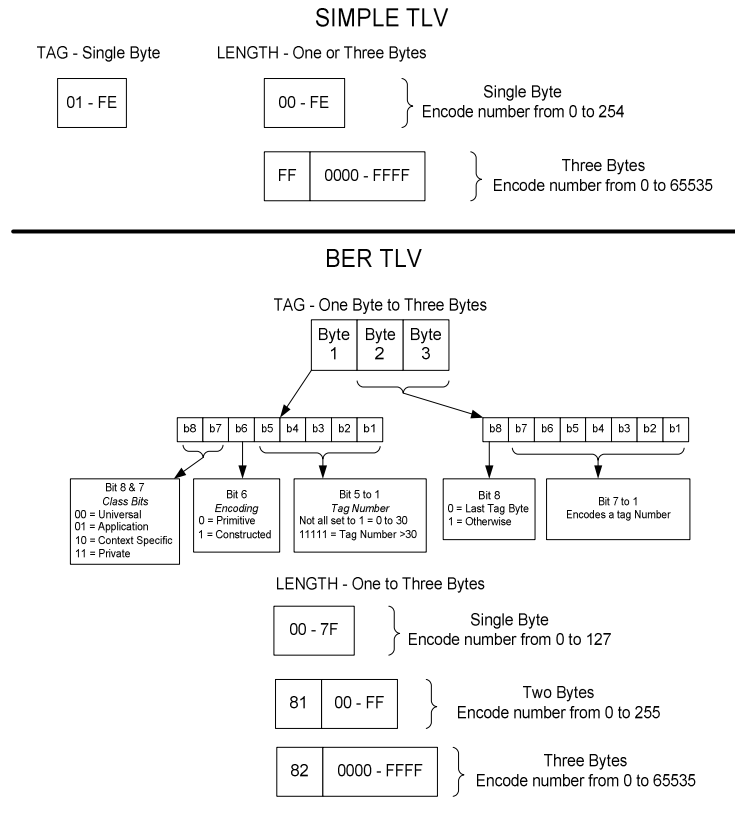
The Security Object format identifies the hash algorithm used when computing the message digests by inclusion of an object identifier. The appropriate object identifiers are identified in the following table.

Table 14 Hash Algorithm Object Identifiers for the 800-73 Security Object

Card Expiration Date	Algorithm
SHA-1	id-sha1 ::= {iso(1)identified-organization(3)oiw(14)secsig(3)algorithms(2)26}
SHA-224	id-sha224 ::= {joint-iso-itu-t(2)country(16)us(840)organization(1)gov(101)csor(3)nistalgorithm(4)hashalgs(2)4}
SHA-256	id-sha224 ::= {joint-iso-itu-t(2)country(16)us(840)organization(1)gov(101)csor(3)nistalgorithm(4)hashalgs(2)1}

Appendix D 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 SIMPLE and BER is shown in the figure below.



Appendix E Standard Biometric Header (CBEFF_HEADER)

- The “PIV” Patron Format CBEFF_HEADER is 88 bytes in length.
- The “PIV” Patron Format CBEFF_HEADER includes several fields not found in CBEFF Patron Format A: BDB Length, SB Length, Federal Agency Smart Credential Number (FASC-N), and Reserved bytes.

Patron Header Version

<i>Length:</i>	1 byte
<i>Data Type:</i>	Unsigned integer
<i>Notes:</i>	The current version is 0x03. This corresponds with the 02/01/2006 publication of SP 800-76.

SBH Security Options

<i>Length:</i>	1 byte
<i>Data Type:</i>	Bitfield
<i>Notes:</i>	

SP 800-76 states that this value should be either 00001101 (signed but not encrypted) or 00001111 (signed and encrypted). Encryption is not permitted when storing minutiae on the PIV card.

The SP 800-76 authors chose to continue a practice from earlier revisions of the document by specifying that the fourth bit is also set. According to INCITS 398-2005 Patron Format “A”, this indicates that the options mask is present in the SBH.

Table A.1 of INCITS 398-2005:

Mask	Note
0x08 (fourth bit)	Optional fields mask present
0x04 (third bit)	Signed
0x02 (second bit)	Privacy (encryption)
0x01 (first bit)	Integrity
0xF0	Reserved. Shall be false.

For Patron Format “PIV”, there is no options mask. So, it must be noted that the fourth bit is set contrary to CBEFF standards. For this reason, the value of this field shall be 0x0D (00001101) when storing minutiae on a PIV card rather than 0x05 (00000101).

BDB Length

<i>Length:</i>	4 bytes
<i>Data Type:</i>	Unsigned integer
<i>Notes:</i>	Contains the length of the Biometric Data Block (BDB).

SB Length

<i>Length:</i>	2 bytes
<i>Data Type:</i>	Unsigned integer
<i>Notes:</i>	Contains the length of the Signature Block (SB).

BDB Format Owner

<i>Length:</i>	2 bytes
<i>Data Type:</i>	Unsigned integer
<i>Notes:</i>	

The format owner for facial images, fingerprint images, and fingerprint minutia is the INCITS Technical Committee on Biometrics (M1). The M1 CBEFF format owner code is 0x001B.

BDB Format Type

<i>Length:</i>	2 bytes
<i>Data Type:</i>	Unsigned integer
<i>Notes:</i>	
Fingerprint Image	0x0401
Fingerprint Minutiae	0x0201
Facial Image	0x0501

For other biometric types, the value shall be determined in accordance with INCITS 398-2005 Section 5.2.1.5.

Biometric Creation Date

<i>Length:</i>	8 bytes
<i>Data Type:</i>	[See Notes]
<i>Notes:</i>	

The date that the biometric was captured (not stored or extracted). The date is to be formatted YYYYMMDDhhmmssZ as 8 bytes in binary representation. "The value for "hh" must be a 24-hour clock value. Each pair of characters is coded in 8 bits. For example, December 15th, 2005 17:35:30 (20051215173530Z) is represented as:

Value	Byte	Hex
20	00010100	0x14
05	00000101	0x05
12	00001100	0x0C
15	00001111	0x0F
17	00010001	0x11
35	00100011	0x23
30	00011110	0x1E
Z (ASCII character for Zulu or Coordinated Universal Time (UTC))	01011010	0x5A

Validity Period

<i>Length:</i>	16 bytes
<i>Data Type:</i>	[See Notes]
<i>Notes:</i>	

These are "not before" and "not after" dates placed in sequence using YYYYMMDDhhmmssZYYYYMMDDhhmmssZ format as described in section 3.2.7.

Biometric Type

<i>Length:</i>	3 bytes
----------------	---------

<i>Data Type:</i>	Unsigned integer	
<i>Notes:</i>		
Fingerprint Image	0x000008	
Fingerprint Minutiae	0x000008	
Facial Image	0x000002	

The fingerprint image and minutiae use the same value as specified in INCITS 398-2005 (CBEFF). The BDB Format Type is used to distinguish between fingerprint minutiae and image. For other biometric types, the value shall be determined in accordance with INCITS 398-2005 Section 5.2.1.5.

Biometric Data Type

<i>Length:</i>	1 byte
<i>Data Type:</i>	Bitfield
<i>Notes:</i>	

Columns one and two of the table below describe the INCITS 398-2005 possible values for this field. The third column notes the examples of applicable biometric data mentioned in SP 800-76.

Data Type	Value	Biometric types
Raw: the data in the BDB is in its raw form as delivered by the sensor.	001xxxxx	Photograph image Fingerprint image
Intermediate: the data in the BDB has been processed from the form delivered by the sensor, but is not in a form usable for matching.	010xxxxx	
Processed: the data in the BDB is in a form that can be used for matching.	100xxxxx	Fingerprint minutia

Biometric Data Quality

<i>Length:</i>	1 byte
<i>Data Type:</i>	Signed integer
<i>Notes:</i>	

For fingerprint images, this is defined as $20(6 - \text{NFIQ})$. No standard is specified for fingerprint minutiae or photographs other than the value must be between -2 and 100. A value of -2 indicates that the measurement is not supported and a value of -1 indicates that an attempt to determine the quality failed.

If multiple biometrics are stored in the Biometric Data Block (BDB), this value is to be the highest quality value of all biometrics found in the BDB.

Photographs shall be coded as -2 to communicate that the quality attribute is not applicable. This is in accordance with INCITS 385-2004.

Creator

<i>Length:</i>	18 bytes
<i>Data Type:</i>	[See Notes]
<i>Notes:</i>	A NULL-terminated string of ASCII characters. There must be at

	least one NULL following the ASCII characters.
--	--

Federal Agency Smart Credential Number (FASC-N)

<i>Length:</i>	25 bytes
<i>Data Type:</i>	[See Notes]
<i>Notes:</i>	The FASC-N component of the Card Holder Unique Identifier (CHUID).

Reserved bytes

<i>Length:</i>	4 bytes
<i>Data Type:</i>	N/A
<i>Notes:</i>	Initialized with NULLS

Multi-byte data stored in the CBEFF_HEADER demands the use of Big Endian ordering. That is, the more significant bytes of any multibyte quantity are stored at lower addresses in memory than (and are transmitted before) less significant bytes. This needs to be carefully tested since many client applications that collect biometrics are based on Little Endian systems.

Creator Construction

Multiple systems will be responsible for the collection of biometrics within the U.S. Department of Defense (DoD). For this reason, the Creator field needs to use a sufficiently unique name so as to distinguish between the various collection systems. All biometrics captured (photograph images, fingerprint images) or generated (fingerprint minutiae) at the CAC issuance station (RAPIDS) shall use the following creator tag where *n*=NULL:

U	S	D	O	D	R	A	P	I	D	S	n	n	n	n	n
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---