NIST Special Publication 800-73-1

Interfaces for Personal Identity Verification

James F. Dray
Scott B. Guthery
Teresa Schwarzhoff

I N F O R M A T I O N S E C U R I T Y

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD, 20899-8930

March 2006

U.S. Department of Commerce
Carlos M. Gutierrez, Secretary

National Institute of Standards and Technology
William Jeffrey, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of non-national security-related information in Federal information systems. This special publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-73-1, 71 pages (March 2006)

Acknowledgements

The authors (James Dray and Terry Schwarhoff of NIST, and Scott Guthery of Mobile Mind, Inc.) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Government Smart Card Interagency Advisory Board (GSC-IAB) and InterNational Committee for Information Technology Standards (INCITS) for providing detailed technical inputs to the SP 800-73 development process. Special recognition is due to Booz Allen Hamilton, and particularly to Ketan Mehta, who made essential technical and editorial contributions. The authors also gratefully acknowledge and appreciate the many contributions from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

Table of Contents

List of Appendices

List of Figures

List of Tables

1. Part 1: Introduction, PIV Data Model and Migration Considerations

The Homeland Security Presidential Directive HSPD-12 called for a common identification standard to be adopted governing the interoperable use of identity credentials to allow physical and logical access to Federal government locations and systems. The Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201 (FIPS 201) [4] was developed to establish standards for identity credentials. This document, Special Publication 800-73 (SP 800-73), specifies interface requirements for retrieving and using the identity credentials from the PIV Card[1] and is a companion document to FIPS 201.

1.1 Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This recommendation is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided A-130, Appendix III.

This recommendation has been prepared for use by federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright though attribution is desirable. Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should this recommendation be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the Office of Management and Budget (OMB), or any other Federal official.

1.2 Purpose

FIPS 201 defines procedures for the PIV lifecycle activities including identity proofing, registration, PIV Card issuance, and PIV Card usage. FIPS 201 also specifies that the identity credentials must be stored on a smart card. This document contains technical specifications to interface with the smart card to retrieve and use the identity credentials. These specifications reflect the design goals of interoperability and PIV Card functions. The goals are addressed by specifying a PIV data model, communication interface, and application programming interface. Moreover, this specification enumerates requirements where the standards include options and branches. This document goes further by constraining implementers' interpretation of the normative standards. Such restrictions are designed to ease implementation, facilitate interoperability, and ensure performance, in a manner tailored for PIV applications.

[1] A physical artifact (e.g., identity card, “smart” card) issued to an individual that contains stored identity credentials (e.g., photograph, cryptographic keys, biometric data) so that the claimed identity of the cardholder can be verified against the stored credentials by another person (human readable and verifiable) or an automated process (computer readable and verifiable).

1.3 Scope

This document specifies the PIV data model, Application Programming Interface (API), and card interface requirements necessary to comply with the mandated use cases, as defined in Section 6 of FIPS 201 and further elaborated in Section 1.7 below, for interoperability across deployments or agencies. Interoperability is defined as the use of PIV identity credentials such that client-application programs, compliant card applications and compliant integrated circuit cards (ICC) can be used interchangeably by all information processing systems across Federal agencies. The specification defines PIV data element identifiers, structure, and format. This specification also describes the client-application programming interface and the card command interface for use of the PIV Card. This document does not address the back-end processes that must be performed to attain full identity assertion.

1.4 Audience and Assumptions

This document is targeted at Federal agencies and implementers of PIV systems. Readers are assumed to have a working knowledge of smart card standards and applications.

1.5 Document Overview

The document describes two realizations of the client-application programming and card command interfaces for personal identity verification: the *transitional* interfaces and the *end-point* interfaces.

The transitional interfaces may be used by agencies with an existing identity card program as an optional intermediate step in evolving to the end-point interfaces. The end-point interfaces are used by agencies without an existing identity card program and by agencies that elect to evolve to the end-point interface in one step rather than two.

The document is divided into three parts as follows:

1.5.1 Part 1: Common Data Model and Migration Considerations

Part 1 consists of Section 1, this first section of the document. In addition to describing the document itself it provides the specification for that which is common to both the transitional and end-point interfaces. Section 1 also includes guidance as to strategies for migrating from the transitional interfaces to the end-point interfaces.

Exactly the same data appears on both the transitional and end-point interfaces. Therefore the description of the data for personal identity verification, the PIV data model, is included in Section 1.

1.5.2 Part 2: The Transitional Interfaces

Part 2 consists of Section 2 of this document. This section describes the subsets of Government Smart Card Interoperability Specification (GSC-IS) [7] that comprise the transitional interface specifications for use by agencies with legacy GSC-IS based card deployments.

1.5.3 Part 3: The End-Point Interfaces

Part 3 consists of Sections 3, 4, 5, 6 and 7 of this document. These sections describe in detail the mandatory end-point PIV card and client application interfaces.

+ Section 3, Concepts and Constructs, describes the model of computation of the PIV client-application programming interface and the PIV Card Application including information processing concepts and data representation constructs.

+ Section 4, Data Objects for Interoperable Use, describes the format and coding of the data structures used by the PIV client-application programming interface and the PIV Card Application.

+ Section 5, Data Types and their Representations, provides the details of the data found on the PIV client-application programming interface and PIV Card Application card command interface.

+ Section 6, The PIV Client-Application Programming Interface, describes the PIV client-application programming interface in programming language independent terms.

+ Section 7, The PIV Card Application Card Command Interface, describes the card command interface to the PIV Card Application.

1.5.4 Appendices

The appendices contain material needing special formatting together with illustrative and exemplar material to aid in understanding information in the body of the document.

1.6 Migration Considerations

This document provides two interface specifications: 1) a transitional Card Specification as described in Part 2; and 2) a FIPS 201 PIV–II Card Specification as described in Part 3. Part 2 is a PIV profile derived from the Government Smart Card Interoperability Specification, Version 2.1(NISTIR 6887). This Part 2 PIV profile is informative, and is presented as one possible path that agencies with existing GSC-IS based smart card deployments may choose to follow during the transition to Part 3 card deployment. All agencies must ultimately comply with Part 3 in accordance with the schedule provided by the Office of Management and Budget (OMB). Full Part 3 deployment is therefore the endpoint of each agency's transition plan.

Agencies may either elect to implement an approved transitional specification (see Section 2) particularly when migrating from currently widely implemented identity card architectures to the Part 3 specifications described in subsequent sections of this publication, or to implement the Part 3 specifications directly. NIST supports agency efforts towards government-wide PIV-II interoperability described in the Part 3 specification. NIST also supports transition specifications for widely implemented deployments as they migrate towards the Part 3 specifications.

The Part 2 migration path is based on continuity of the PIV data model. Specific considerations associated with this migration path are highlighted below:

+ Part 2 presents a subset of the dual GSC-IS card edge interfaces. Part 3 presents a unified card edge interface that is technology independent and compliant with existing international standards.

+ Part 3 provides limited credential administration functionality. A unified and interoperable card management solution between issuing domains including the loading of new card applications is not provided.

+ Named data objects within the data model may be directly accessed. If a data object is managed by the default application, it can be retrieved directly without selecting the application. This

avoids a requirement to search through discovery to get named data objects. Otherwise, the (non-default) application managing the data object is selected and the data object is retrieved from this application. The GET DATA command described in Section 6 retrieves a data object in one command.

+ The data model including the data model namespace is controlled by NIST and hence change management of well known and interoperable data objects will be managed by NIST in the process of managing the overall data model. As a first step in namespace management, the data object identifiers of GSC-IS and transitional systems in the range '0000' through '9FFF' will be explicitly managed by NIST and data object identifiers of GSC-IS and transitional systems in the range 'A000' through 'FFFF' are placed under control of the card issuer.

+ Each application managing one or more of the directly addressable data model data objects will have a version number enabling the relying application to figure out the level of the information contained within the object. The version of the Part 3 PIV Card Application is encoded in its full Application IDentifier (AID) which is returned when this application is selected. This is in addition to the Card Capability Container (CCC) style data model naming facility carried over from GSC-IS.

+ Agency-specific applications can be included on cards containing PIV applications. These applications may define and manage their own namespaces that are used when the application is used. Such applications will have application identifiers outside the application namespace managed by NIST; that is, application identifiers not rooted on the NIST Registered application provider IDentifier (RID).

1.7 PIV Data Model

The PIV data model for SP 800-73 is constructed according to GSC-IS specifications. Table 1 defines a high level view of the data model. Each container is labeled either as Mandatory or Optional. Mandatory data elements are common to both Part 2 and 3. This data model is designed to enable and support dual interface cards. Note that access conditions based on the interface mode (contact vs. contactless) take precedence over all Access Rules defined in Table 1, Column 3.

Appendix A provides a detailed spreadsheet for the data model. ContainerIDs and Tags within the containers are defined by this data model and in accord with SP 800-73 naming conventions. It provides guidance on lengths for fields and sizes for buffers. These lengths and sizes are under issuer control, as are optional containers. Issuers should calculate their specific data size requirements for implementation specific needs.

Table 1. SP 800-73 Data Model Containers

RID 'A0 00 00 00 01 16'	ContainerID	Access Rule	Contact / Contactless	M/O
Card Capability Container	0xDB00	Read Always	Contact	Mandatory
CHUID Buffer	0x3000	Read Always	Contact & Contactless	Mandatory
PIV Authentication Certificate Buffer	0x0101	Read Always	Contact	Mandatory
Fingerprint Buffer	0x6010	PIN	Contact	Mandatory
Printed Information Buffer	0x3001	PIN	Contact	Optional
Facial Image Buffer	0x6030	PIN	Contact	Optional
Digital Signature Certificate Buffer	0x0100	Read Always	Contact	Optional
Key Management Certificate Buffer	0x0102	Read Always	Contact	Optional
Card Authentication Certificate Buffer	0x0500	Read Always	Contact	Optional
Security Object Buffer	0x9000	Read Always	Contact	Mandatory

1.8 Mandatory Data Elements

The mandatory data containers support FIPS 201 minimum mandatory compliance.

1.8.1 Card Capability Container

The CCC is mandatory for compliance with the GSC-IS specification. It supports minimum capabilities for lookup on data model and application information.

The data model shall be identified by data model number “0x10”. Deployed applications use “0x00” through “0x04”. This enables the GSC-IS application domain to correctly identify a new data model name space and structure as defined in this document.

1.8.2 PIV Authentication Key

The PIV Authentication Key as defined in FIPS 201 is used to authenticate the card and cardholder using the Personal Identification Number (PIN).

1.8.3 CHUID

The Cardholder Unique Identifier (CHUID) buffer is defined in accordance with the Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems (TIG SCEPACS). [5] For this specification, the CHUID is common between the contact and contactless chips. For dual chip implementations, the CHUID is copied in its entirety between the two chips.

In addition to the requirements specified in TIG SCEPACS, the CHUID on a PIV Card shall meet the following requirements:

+ 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 9,999,999,999 credentials.

+ The Global Unique Identifier (GUID) field must be present, and may include either an issuer assigned IPv6 address or be coded as all zeros. The GUID is included to enable future migration away from the FASC-N into a robust numbering scheme for all issued credentials.

+ The DUNS and Organizational Code fields are optional.

+ The Authentication Key Map is specified as an optional field which enables the application to discover the key reference. This is one method of implementing the symmetric challenge/response protocols using the Card Authentication Key.

+ The Expiration Date is mapped to the reserved for future use (RFU) tag 0x35, keeping that within the existing scope of the TIG SCEPACS specification. This field shall be 8 bytes in length and shall be encoded as YYYYMMDD.

+ The CHUID is signed in accordance with FIPS 201. The card issuer's digital signature key shall be used to sign the CHUID. The signature field of the CHUID contains the card issuer's certificate.

1.8.4 Fingerprints

The fingerprint buffers specify the primary and secondary fingerprints in accordance with the FIPS 201. The Common Biometric Exchange Formats Framework (CBEFF) headers shall contain the FASC-N and shall require the Integrity Option. The headers shall not require the Confidentiality Option.

1.8.5 Security Object

The security object is in accordance with Appendix C of PKI for Machine Readable Travel Documents Offering ICC Read-Only Access Version 1.1. [8] Tag "0xBA" is used to map the ContainerIDs in the PIV data model to the 16 Data Groups specified in the Machine Readable Travel Document (MRTD). This enables the security object to be fully compliant for future activities with identity documents.

The card issuer's digital signature key used to sign the CHUID shall also be used to sign the security object. The signature field of the security object shall omit the issuer's certificate, since it is included in the CHUID.

1.9 Optional Data Elements

The optional data elements of FIPS 201, when implemented, shall conform to the specifications provided in this document.

1.9.1 Printed Information Buffer

All FIPS 201 mandatory information printed on the card is duplicated on the chip in this buffer. The Security Object enforces integrity of this information according to the issuer. This provides specific protection that the card information must match the printed information, mitigating alteration risks on the printed media.

1.9.2 Facial Image Buffer

The photo on the chip supports human verification only. It is not intended to support facial recognition systems for automated identity verification. The Security Object enforces integrity of this information

according to the issuer. This provides specific protection that the card information must match the printed information, mitigating alteration risks on the printed media.

1.9.3 Digital Signature Key

This key and certificate supports the use of digital signatures for the purpose of document signing. The Public Key Infrastructure (PKI) cryptographic function is protected with a "PIN Always" access rule. This requires cardholder participation every time the key is used for digital signature generation.

1.9.4 Key Management Key

This key and certificate supports the use of encryption for the purpose of confidentiality. This key pair is escrowed by the issuer for key recovery purposes. The PKI cryptographic function is protected with a "PIN" access rule. This requires cardholder activation, but enables multiple compute operations without additional cardholder consent.

1.9.5 Card Authentication Key

This key and certificate if the key is an asymmetric key supports PIV Card Authentication for device to device authentication purposes. Cardholder consent is not required to use this key. The access rule for PKI cryptographic functions is "Always". Where the Card Authentication Key is a symmetric key, the CHUID authentication key map shall be present and specify the cryptographic algorithm and key storage location.

2. Part 2: Transition Card Interfaces

2.1 PIV Application Programming Interface

2.1.1 Basic Services Interface

This chapter defines the Basic Services Interface (BSI) services provided to a PIV application. The following specification is compliant with NIST Interagency Report (NISTIR) 6887 unless otherwise specified.

The functions listed hereafter are a subset of NISTIR 6887; these functions 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()`

2.2 PIV Card Application Version

The application hosting the CHUID is always mandatory on the PIV Card in BOTH contact and contactless modes. When selected, that application must return the PIV Application version on response to select.

The PIV Application Version indicates:

1. Reference to the supported card edge specifications
2. PIV Data Model Object Identifier
3. List of mandatory applications with their AIDs

4. Reference to the mandatory subset of the PIV data model for each application (Object Identifiers)
5. Cryptographic capabilities for each application

Specific values are specified in both contact and contactless modes.

The PIV Application Version is returned on response to SELECT to the PIV Application containing the CHUID on the card, for both contact and contactless modes.

The last byte of the application name returned indicates the Application version in the card and the card type of Virtual Machine (VM) or File System (FS).

The Application version byte returned by the card is structured as follow:

Bit8 = 0b VM card edge
Bit8 =1b FS card edge

Bits 7-1 PIV application version in this given card
This number indicates the release of the SP800-73 specification that the PIV Card is following.

Table 2. Full PIV Card Versions

PIV Application Version	Description	Data Model
0x00	FULL PIV on VM Card	PIV Data Model Object Identifier
0x80	FULL PIV on FS Card	PIV Data Model Object Identifier

2.2.1 PIV Objects Naming Structure

+ At the card edge level, the Objects are referenced by GSC-IS Object Identifier (2 bytes), and are located within an AID. Where AID = Issuer RID (5bytes) || PIX. In this context, the Proprietary Identifier eXtension (PIX) consists of 2 bytes: Application ID.

+ Each card application Uniform Resource Locator (URL) listed in the CCC consists of the following sequence of elements:

    - Issuer RID (any value assigned to the card issuer)
    - Card Application Type: PKI, GC: indicates the Application Protocol Data Unit (APDU) commands available on that object.
    - GSC-IS Object Identifier: Identifies the Container or Object to Select.
    - AID or PIX.

All the following CardApplicationURL fields (AccessProfile, pinID, AccessKeyInfo, keyCryptoAlgorithm) are not present in the context of the PIV Application on VM Cards, but are optional on File System Cards.

+ At the BSI level, the objects are referenced by 7 bytes GSC-IS Object AID (Issuer RID || GSC-IS Object Identifier).

+ At a door reader or when PIV applications directly access the card edge, the objects are referenced by 2 bytes GSC-IS Object Identifiers.

2.2.2 Mapping mechanisms

The CCC CardApplicationURL is used to lookup the GSC-IS Object Identifier and construct the corresponding application identifier for selection.

2.3 Card Edge Commands

2.3.1 General

The PIV application supports a dual VM and File System card edge to assure interoperability and maintain compatibility with existing GSC-IS based systems.

The PIV Application also requires a contactless interface. The contactless command interface is compliant with NISTIR 6887 Appendix G. However, in both cases (Virtual Machine and File System), the contactless interface relies on the data model Object IDs and Tags defined in this 800-73 specification rather than the Object IDs and Tags defined in Appendix G. 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.

The information presented at the interface has a format that is specific to the card edge type as described in NISTIR 6887.

2.3.2 Data Format and Structure

See NISTIR 6887 Sections 8.2, 8.3, 8.4.

2.3.3 PIV Card Edge Commands

To satisfy the requirements of PIV, only a subset of the GSC-IS commands are required. The APDUs are divided into two categories: Commands for Common Interface and Commands for Authentication.

Note: PIV Cards must support either the VM Card edge or the FS card edge. A mix and match of APDUs between card edges is not allowed.

The ADPU commands and responses are defined in NISTIR 6887, Table 5-2 and 5-3.

To implement a PIV Card using VM card commands, the following card commands are needed.

Table 3. VM Card Commands

Type	Name
Commands for common interface	SELECT APPLET / SELECT OBJECT
	GET RESPONSE
Card platform commands for common interface	READ BUFFER
Commands for authentication	VERIFY
	PRIVATE SIGN / DECRYPT

Note that the usable command set depends on the currently selected object.

+ After a selection of a container object, all commands above but PRIVATE SIGN/DECRYPT are available.

+ After a selection of a PKI object, all commands above are available.

To implement a PIV Card using file system card commands, the following card commands are needed.

Table 4. File Card Commands

Type	Name
Commands for common interface	SELECT
	GET RESPONSE
Card Platform Commands for Common Interface	READ BINARY
Commands for Authentication	VERIFY
	MANAGE SECURITY ENVIRONMENT
	PERFORM SECURITY OPERATION

2.3.3.1 VM Card Platform Commands for Common Interface

2.3.3.1.1 SELECT APPLET/SELECT OBJECT APDU

The SELECT command serves two purposes in a VM card 1) sets the currently selected application 2) sets the currently selected object.

Command Message

CLA	0x00
INS	0xA4
P1	Reference Control Parameter
P2	0x00
Lc	Length of the Data field
Data Field	Applet AID or Card Object ID
Le	Empty

Reference control parameter P1

Parameter P1 indicates the type of selection to perform. The accepted values are:

- 04h for selecting an application by AID and as a consequence selection of the default object in this application.
- 02h for selecting an object by Object Identifier.
- This command supports selection of the object using AID or File ID.

Data field sent in the command message

In the case of application selection, the data field contains the AID.

In the case of object selection, the data field contains the Object Identifier

Response Message

Data field returned in the response message

If the APDU result indicates success,

For selecting a card object, the response message is null but Status Word (SW);

Addition to NISTIR 6887: For selecting an applet, the response message contains the minimum File Control Information defined in ISO-7816-4 (FCI), as follows:

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

Processing state returned in the response message

SW1	SW2	Meaning
6A	82	Application not found
90	00	Successful Execution
69	99	Select Fails (returned by card platform) - addition to NISTIR 6887

2.3.3.1.2 GET RESPONSE APDU

See NISTIR 6887, Section 5.3.3.6.

2.3.3.1.3 READ BUFFER APDU

See NISTIR 6887, Section 5.3.4.2.

2.3.3.2 VM Card Platform Commands for Authentication

2.3.3.2.1 VERIFY APDU

This APDU is used to compare the PIN with corresponding authentication data on the smart card. The host sends the authentication data in this APDU and directs the smart card to compare it with authentication data on the smart card. The authentication data is passed unencrypted.

Command Message

CLA	0x00
INS	0x20
P1	0x00
P2	0x00
L_c	Length of Data Field. Must be 8
Data Field	Authentication data (i.e., PIN)
L_e	Empty

Note: If Lc=0x00 and the command data field is empty, the command can be used to retrieve the number of further retries allowed or to check whether verification is not needed.

Key Reference Identifier P2

In addition to NISTR 6887 the PIN used in PIV Cards using the File Card Edge shall comply with the PIN format defined in Section 3.5.3.

Response Message

Data Field returned in the Response Message

Empty.

Processing State returned in the Response Message

SW1	SW2	Meaning
63	00	Verification failed
63	CX	Verification failed, X indicates the number of further allowed retries
69	83	Authentication method blocked
69	84	Referenced data deactivated
6A	86	Incorrect parameters P1-P2
6A	88	Reference data not found
90	00	Correct execution

2.3.3.2.2 PRIVATE SIGN/DECRYPT APDU

This command is used to perform an Rivest, Shamir, Aldeman (RSA) signature or data decryption.

Command Message

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

Reference control parameter P1 (Addition to NISTIR 6887)

Control parameter P1 indicates whether more blocks containing the data follows. This is used to chain multiple APDUs in order to transport the input data for 2048-bit or greater RSA operations. Note that this command chaining method is not compliant with international integrated circuit standards ISO/IEC 7816, Information Technology – Identification Cards – Integrated Circuit(s) Card with Contacts, [1] nor therefore is it compatible with the Part 3 command chaining method which is compliant with international ICC standards.

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	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

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

SW1	SW2	Meaning
67	00	Command data length not equal to RSA key size
69	83	RSA Private Key not initialized
69	82	Access condition not satisfied
69	85	Conditions of use not satisfied (Current selected object is not valid)
61	XX	for “normal processing, XX bytes of data is read and available for a subsequent Get Response”

2.3.3.3 File Card Platform Commands

2.3.3.3.1 SELECT APDU

See NISTIR 6887 sections 5.1.1.3, 5.1.1.3.1, 5.1.1.4 and 5.1.1.5. Only two SELECT commands are required for the transitional PIV application. "Select EF" is required to be supported (P1 = 0x00 or P1 0x02) and "Select File by name" is required to be used to access the CCC for all cards. This departs from GSC-IS which uses a different mechanism to find out about the card edge available on the card.

Note: in GSC-IS, even if a File Card supports the APDU command Select DF by name, the CCC shall not use the GSA AID (A00000000116DB00) as the name of the CCC allowing to find out it is a file card. This is different in this specification.

2.3.3.3.2 GET RESPONSE APDU

See NISTIR 6887, Section 5.1.1

2.3.3.3.3 READ BINARY APDU

See NISTIR 6887, Section 5.1.1

2.3.3.3.4 MANAGE SECURITY ENVIRONMENT

See NISTIR 6887 Section 5.1.3.1

2.3.3.3.5 PERFORM SECURITY OPERATION APDU

See NISTIR 6887, Section 5.1.1

2.3.3.3.6 VERIFY APDU

See NISTIR 6887, Section 5.1.1

In addition to NISTR 6887 the PIN used in PIV Cards using the File Card Edge shall comply with the PIN format defined in Section 3.5.3.

2.4 General Status Conditions

See NISTIR 6887 for General Status Conditions.

3. Part 3: End-Point Concepts and Constructs

Special Publication 800-73 Part 3 defines two interfaces to an ICC that contains the Personal Identity Verification card application: a high-level PIV client-API and a low-level PIV Card Application card command interface (card edge).

The information processing concepts and data constructs on both interfaces are identical and may be referred to generically as the information processing concepts and data constructs on the *PIV interfaces* without specific reference to the client-application programming interface or the card command interface.

The client-application programming interface provides task-specific programmatic access to these concepts and constructs and the card command interface provides communication access to concepts and constructs. The client-application programming interface is used by client applications using the PIV Card Application. The card command interface is used by software implementing the client-application programming interface (middleware).

The client-application programming interface is thought of as being at a higher level than the card command interface because access to a single entry point on the client-application programming interface may cause multiple card commands to traverse the card command interface. In other words, it may require more than one card command on the card command interface to accomplish the task represented by a single call on an entry point client-application programming interface

The client-application programming interface is a program execution, call/return style interface whereas the card command interface is a communication protocol, command/response style interface. Because of this difference the representation of the PIV concepts and constructs as bits and bytes on the client-application program interface may be different from the representation of these same concepts and constructs on the card command interface.

3.1 Unified Card Command Interface

The card command interface of the PIV Card Application is a unification of the two card command interfaces found in GSC-IS and in Section 2 described above.

This unification is accomplished by adopting the object-oriented model of computation of the GSC-IS virtual machine card edge and realizing its technical details using the data structures and operations found in the international ICC standards underpinning the GSC-IS file system card edge [1]. This brings the PIV Card Application into conformance with those standards with minimal impact on existing GSC-IS deployments.

As a result of this unification, the behavior of the PIV Card Application and the client-applications accessing it is independent of the ICC platform on which the PIV Card Application is installed.

3.1.1 Platform Requirements

The following are the requirements that the PIV Card Application places on the ICC platform on which it is implemented or installed:

- global security status that includes the security status of a global cardholder PIN
- application selection using a truncated AID
- ability to reset the security status of an individual application

- indication to applications as to which physical communication interface – contact versus contactless – is in use
- support for the default selection of an application upon warm or cold reset.

3.2 Namespaces of the PIV Card Application

Names used on the PIV interfaces are drawn from three namespaces managed by NIST:

- PIXes of the NIST RID
- ASN.1 object identifiers (OIDs) in the personal verification subset of the OIDs managed by NIST
- Basic Encoding Rules – Tag Length Value (BER-TLV) tags of the NIST PIV coexistent tag allocation scheme

All unspecified names in these managed namespaces are reserved for future use.

All interindustry tags defined in [1] and used in the NIST coexistent tag allocation scheme without redefinition have the same meaning in the NIST PIV coexistent tag allocation scheme as they have in [1].

All unspecified values in the following identifier and value namespaces are reserved for future use:

- algorithm identifiers
- key reference values
- cryptographic mechanism identifiers

3.3 Data Objects

A *data object* is an item of information seen on the card command interface for which are specified a name, a description of logical content, a format and a coding. Each data object has a globally unique name called its *object identifier* as defined in ISO/IEC 8824-2:2002, Information technology – Abstract Syntax Notation One (ASN.1): Information object specification. [2]

A data object whose data content is encoded as a BER-TLV data structure as in ISO/IEC 8825-1:2002, Information technology – ASN.1 encoding rules, [3] is called *BER-TLV data object.*

3.3.1 Data Object Content

The *content* of a data object is the sequence of bytes that are said to be *contained in* or to be the *value of* the data object. The number of bytes in this byte sequence is referred to as the *length* of the data content and also as the *size* of the data object. The first byte in the sequence is regarded as being at *byte position* or *offset* zero in the content of the data object.

The data content of a BER-TLV data object may consist of other BER-TLV data objects. In this case the tag of the data object indicates that data object is a *constructed data object.* A BER-TLV data object that is not a constructed data object is called a *primitive data object*.

3.4 Card Applications

Each command that appears on the card command interface shall be implemented by a *card application* that is resident in the ICC. The card command enables one to perform operations on and with the data objects to which the card application has access.

Each card application shall have a globally unique name called its AID [1, Part 4]. Access to the card commands and data objects of a card application shall be gained by selecting the card application using its application identifier. The PIX of the AID shall contain an encoding of the version of the card application.

The card application whose commands are currently being used is called the *currently selected application*.

3.4.1 Personal Identity Verification Card Application

The AID of the Personal Identity Verification card application (PIV Card Application) shall be:

'A0 00 00 03 08 00 00 10 00 01 00'

The AID of the PIV Card Application consists of the NIST RID ('A0 00 00 03 08') followed by the application portion of the NIST PIX indicating the PIV Card Application ('00 00 10 00') and then the version portion of the NIST PIX ('01 00') for the first version of the PIV Card Application. All other PIX sequences on the NIST RID including the trailing five bytes PIV Card Application AID are reserved for future use.

The PIV Card Application surfaces the card commands described in Section 7 on the card command interface.

3.4.2 Default Selected Card Application

The card platform shall support a default selected card application. In other words, there shall be a currently selected application immediately after a cold or warm reset. This card application is the default selected card application.

Since the PIV Card Application can be selected by truncated AID, the selected card application may be the PIV Card Application or it may be another card application.

3.5 Security Architecture

The security architecture of an ICC is the means by which the security policies governing access to each data object stored on the card are represented within the card.

The software in the ICC applies these security policy representations to all card commands thereby ensuring that the prescribed data policies for the card applications are enforced.

The following subsections describe the security architecture of the PIV Card Application.

3.5.1 Access Control Rule

An *access control rule* shall consist of an *access mode* and a *security condition*. The access mode is an operation that can be performed on a data object. A security condition is a Boolean expression using variables called security statuses that are defined below.

According to an access control rule, the action described by the access mode can be performed on the data object if and only if the security condition evaluates to TRUE for the current values of the security

statuses. If there is no access control rule with an access mode describing a particular action, then that action shall never be performed on the data object.

3.5.2 Security Status

Associated with each authenticatable entity shall be a set of one or more Boolean variables each called a *security status indicator* of the authenticatable entity. The security status indicator of an authenticatable entity shall be TRUE if the credentials associated with the security status indicator of the authenticatable entity have been authenticated and FALSE otherwise.

The successful execution of an authentication protocol shall set the security status indicator associated with the credentials that were verified by the protocol to TRUE.

As an example, the credentials associated with three security status indicators of the card holder might be: PIN, fingerprint, and voice biometric. Demonstration of knowledge of the PIN is the authentication protocol for the first security status indicator. Comparison of the fingerprint template on the card with a fingerprint acquired from the card holder is the authentication protocol for the second security status indicator. Acquisition of a voice sample and comparison with a voice template is the authentication protocol for the third. A security condition using these three security status indicators might be ((PIN AND fingerprint) OR (voice biometric)).

A security status indicator shall be said to be a *global* security status indicator if it is not changed when the currently selected application changes from one application to another.

A security status indicator is said to be an *application* security status indicator if it is set to FALSE when the currently selected application changes from one application to another. Every security status indicator is either a global security status indicator or an application security status indicator.

The term *global security status* refers to the set of all global security status indicators. The term *application security status* refers to the set of all application security status indicators for a specific application.

3.5.3 Authentication of an Individual

Knowledge of a PIN is one means by which an individual can be authenticated to the PIV Card Application.

Personal identification numbers presented to the card command interface shall be 8 bytes long. If the actual PIN length is less than 8 bytes it shall be padded to 8 bytes with ‘FF’. The 'FF' padding bytes shall be appended to the actual PIN. The bytes comprising the PIN shall not include 'FF'. For example,

- Actual PIN: “123456” or ‘31 32 33 34 35 36’
- Padded PIN presented to the card command interface: ‘31 32 33 34 35 36 FF FF’

3.6 Current State of the PIV Card Application

The elements of the *current state* of the PIV Card Application when the PIV Card Application is the currently selected application are described in Table 5.

Table 5. State of the PIV Card Application

State Name	Always Defined	Comment	Location of State
Global security status	Yes	Contains security status indicators that span all card applications on the platform.	PIV Platform
Currently selected application	Yes	The platform shall support the selection of a card application using a possibly right-truncated application identifier and there shall always be a currently selected application.	PIV Platform
Application security status	Yes	Contains security status indicators local to the PIV Card Application.	PIV Card Application

4. Part 3: End-Point Data Objects

4.1 PIV Card Application Data Objects

A PIV Card Application shall contain six mandatory data objects and five optional data object for interoperable use. The six mandatory data objects for interoperable use are as follows:

1. Card Capability Container
2. Card Holder Unique Identifier
3. X.509 Certificate for PIV Authentication
4. Card Holder Fingerprint I
5. Card Holder Fingerprint II[2]
6. Security Object

The five optional data objects for interoperable use are as follows:

1. Card Holder Facial Image
2. Printed Information
3. X.509 Certificate for PIV Digital Signature
4. X.509 Certificate for PIV Key Management
5. X.509 Certificate for Card Authentication

4.2 OIDs and Tags of PIV Card Application Data Objects

Table 6 lists the ASN.1 object identifiers and BER-TLV tags of the eleven PIV Card Application data objects for interoperable use. For the purpose of constructing PIV Card Application data object names in the CardApplicationURL in CCC of the PIV Card Application, the NIST RID ('A0 00 00 03 08') shall be used and the card application type shall be set to '00'. The last byte of the three-byte BER-TLV tag is equivalent to a container ID for purposes of constructing the Security Object. Table 1 lists the access control rules of the eleven PIV Card Application data objects for interoperable use. See Table 12 for the key references and algorithms associated with these authenticatable entities.

Table 6. Object Identifiers of the PIV Data Objects for Interoperable Use

Data Object for Interoperable Use	ASN.1 OID	BER-TLV Tag	M/O
Card Capability Container	2.16.840.1.101.3.7.1.219.0	'5FC107'	M
Card Holder Unique Identifier	2.16.840.1.101.3.7.2.48.0	'5FC102'	M
X.509 Certificate for PIV Authentication	2.16.840.1.101.3.7.2.1.1	'5FC105'	M
Card Holder Fingerprints	2.16.840.1.101.3.7.2.96.16	'5FC103'	M
Printed Information	2.16.840.1.101.3.7.2.48.1	'5FC109'	O
Card Holder Facial Image	2.16.840.1.101.3.7.2.96.48	'5FC108'	O
X.509 Certificate for Digital Signature	2.16.840.1.101.3.7.2.1.0	'5FC10A'	O
X.509 Certificate for Key Management	2.16.840.1.101.3.7.2.1.2	'5FC10B'	O
X.509 Certificate for Card Authentication	2.16.840.1.101.3.7.2.5.0	'5FC101'	O

[2] Note that both Card Holder Fingerprints will be recorded within one container on the PIV Card.

Data Object for Interoperable Use	ASN.1 OID	BER-TLV Tag	M/O
Security Object	2.16.840.1.101.3.7.2.144.0	‘5FC106’	M

5. Part 3: End-Point Data Types and Their Representations

This section provides a description of each data type found on the PIV client-application programming and PIV Card Application command interfaces. Unless otherwise indicated the representation shall be the same on both interfaces.

5.1 Algorithm Identifier

An algorithm identifier shall be a one-byte identifier of a cryptographic algorithm together with a mode of operation and reference data length. Table 7 lists the algorithm identifiers for the cryptographic algorithms that may be recognized on the PIV interfaces. All other algorithm identifier values are reserved for future use.

Table 7. Cryptographic Algorithm Identifiers

Algorithm Identifier	Algorithm – Mode	Reference Data Length	M/O
'01'	2 Key Triple DES – ECB	128 bits	O
'02'	2 Key Triple DES – CBC	128 bits	O
'03'	3 Key Triple DES – ECB	192 bits (including parity bits)	M
'04'	3 Key Triple DES – CBC	192 bits (including parity bits)	M
'05'	RSA	3072 bits	O
'06'	RSA	1024 bits	M
'07'	RSA	2048 bits	O
'08'	AES-128 – ECB	24 bytes	O
'09'	AES-128 – CBC	24 bytes	O
'0A'	AES-192 – ECB	36 bytes	O
'0B'	AES-192 – CBC	36 bytes	O
'0C'	AES-256 – ECB	48 bytes	O
'0D'	AES-256 – CBC	48 bytes	O
'0E'	ECC: Curve P-224	224 bits	O
'0F'	ECC: Curve K-233	233 bits	O
'10'	ECC: Curve B-233	233 bits	O
'11'	ECC: Curve P-256	256 bits	O
'12'	ECC: Curve K-283	283 bits	O
'13'	ECC: Curve B-283	283 bits	O

The default cryptographic algorithm for the PIV Card Application with algorithm identifier '00' is 3 Key Triple DES – ECB. A technical note synchronizing SP 800-73 and the forthcoming SP 800-78, Cryptographic Algorithms and Key Sizes for Personal Identity Verification, on message padding will be issued in the future.

5.2 Application Property Template

Upon selection, the PIV Card Application shall return the application property template described in Table 8.

Table 8. Data Objects in the PIV Card Application Property Template (Tag '61')

Description	Tag	M/O	Comment
Application identifier of application	'4F'	M	The PIX of the AID includes the encoding of the version of the PIV Card Application.
Coexistent tag allocation authority	'79'	M	Coexistent tag allocation authority template. See Table 9.
Application label	'50'	O	Text describing the application; e.g. for use on a man-machine interface.
Uniform resource locator	'5F50'	O	Reference to the specification describing the application.

Table 9. Data Objects in a Coexistent Tag Allocation Authority Template (Tag '79')

Description	Tag	M/O	Comment
Application identifier	'4F'	M	NIST will post the PIV RID on http:///csrc.nist.gov/piv-project and will publish it in a technical note.

5.3 Authenticator

The authenticator BER-TLV used on the PIV client-application programming interface shall have the structure described in Table 10.

Table 10. Data Objects in an Authenticator Template (Tag '67')

Description	Tag	M/O	Comment
Reference data	'81'	M	E.g. the PIN value or challenge response
Key reference	'83'	M	See Table 12.

5.4 Connection Description

The connection description BER-TLV used on the PIV client-application programming interface shall have the structure described in Table 11.

Table 11. Data Objects in a Connection Description Template (Tag '7F21')

Description	Tag	M/O	Comment
Interface device – PC/SC	'81'	C	Card reader name
Interface device – SCP	'82'	C	Card reader identifier on terminal equipment
Interface device – EMR	'83'	C	Contactless connection using radio transmission
Interface device – IR	'84'	C	Contactless connection using infrared transmission
Interface device – PKCS#11	'85'	C	PKCS#11 interface
Interface device – CryptoAPI	'86'	C	CryptoAPI interface
Network node – Local	'90'	C	No network between client-application host and card reader host
Network node – IP	'91'	C	IP address of card reader host
Network node – DNS	'92'	C	Internet domain name of card reader host
Network node – ISDN	'93'	C	ISDN dialing number string of terminal equipment containing the card reader

At most one selection from the '8x' series and one selection from the '9x' series shall appear in the connection description template.

For example, '7F 21 0C 82 04 41 63 6D 65 91 04 81 06 0D 17' describes a connection to a generic card reader at Internet address 129.6.13.23. As another example, '7F 21 0B 82 01 00 93 06 16 17 12 34 56 7F' describes a connection to the subscriber identity module in the mobile phone at +1 617 123 4567.

When used as an argument to the pivConnect entry point on the PIV client-application programming described in Section 5.1.1, an '8x' series data object with zero length together with a '9x' series data object request the return of all available card readers of the described type on the described node. Thus, '7F 21 04 81 00 90 00' would requests a list of all available PC/SC card readers on the host on which the client-application was running.

5.5 Key References

A key reference is a 6-bit identifier of cryptographic material in the PIV Card Application used in a cryptographic protocol. When represented as a byte, the key reference occupies b8 and b5-b1 while b7 and b6 shall be set to 0. If b8 is 0 then the key reference names global reference data. If b8 is 1 then the key reference names application-specific reference data.

Table 12 defines the key reference values that shall be used on the PIV interfaces. Key references are only assigned to private and secret (symmetric) keys. All other PIV Card Application key reference values are reserved for future use.

Table 12. PIV Card Application Authentication Algorithms and Key References

Algorithm Identifier	Key Reference Value	Key Reference Name	Authenticatable Entity	Security Condition for Use	Retry Reset Value	Number of Unblocks
N/A	'00'	Global PIN	Card Holder	Always	Platform Specific	Platform Specific
N/A	'80'	Application PIN	Card Holder	Always	Issuer Specific	Issuer Specific
'06'	'9A'	PIV Authentication Key	PIV Card Application Provider	PIN	N/A	N/A
'00'	'9B'	PIV Card Application Administration Key	PIV Card Application Administrator	Always	N/A	N/A
'06'	'9C'	PIV Card Application Digital Signature Key	PIV Card Application Administrator	PIN / Always	N/A	N/A
'06'	'9D'	PIV Card Application Key Management Key	PIV Card Application Administrator	PIN	N/A	N/A

The card holder global PIN may be referenced in PIV Card Application access control rules but its current status shall not be changed, its value shall not be changed nor shall its retry counter be reset while the PIV Card Application is the currently selected application.

5.6 Status Words

A status word shall be a 2-byte value returned by an entry point on the client-application programming interface or a card command at the card edge. The first byte of a status word is referred to as SW1 and the second byte of a status word is referred to as SW2.

Recognized values of all SW1-SW2 pairs used as return values on both the client-application programming and card command interfaces and their interpretation are given in Table 13. The description of individual client-application programming interface entry points or card commands provide additional information for interpreting the status words they return.

Table 13. Status Words

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available

SW1	SW2	Meaning
‘62’	‘82’	End of data encountered
‘63’	‘xx’	Warning; see entry point or command for specifics
'63'	'CX'	Verification failed, X indicates the number of further allowed retries or resets
‘68’	‘xx’	Communication error; see entry point or command for specifics
‘69’	‘82’	Security condition not satisfied
‘69’	‘83’	Authentication method blocked
‘69’	‘85’	Condition of use not satisfied
‘6A’	‘80’	Incorrect parameter in command data field
‘6A’	‘81’	Function not supported
'6A'	'84'	Not enough memory
‘6A’	‘86’	Incorrect parameter in P1 or P2
‘6A’	‘88’	Referenced data or reference data not found
'90'	'00'	Successful execution

5.7 Object Identifiers

Each of the data objects in the PIV Card Application has been provided with an ASN.1 OID from the NIST personal verification arc and a three-byte BER-TLV tag. These object identifier assignments are given in Table 6.

A data object shall be identified on the PIV client-application programming interface using its OID. An object identifier on the PIV client-application programming interface shall be a dot delimited string of the integer components of the OID. For example, the representation of the OID of the CHUID on the PIV client-application programming interface is “2.16.840.1.101.3.7.2.48.0”.

A data object shall be identified on the PIV Card Application card command interface using its BER-TLV tag. For example, the CHUID is identified on the card command interface to the PIV Card Application by the three-byte identifier ‘5FC102’.

6. Part 3: End-Point Client-Application Programming Interface

Table 14 lists the entry points on the PIV client-application programming interface.

Table 14. Entry Points on PIV Client-Application Programming Interface

Type	Name
Entry Points for Communication	**pivConnect**
	pivDisconnect
Entry Points for Data Access	**pivSelectCardApplication**
	pivLogIntoCardApplication
	pivGetData
	pivLogoutOfCardApplication
Entry Points for Cryptographic Operations	**pivCrypt**
Entry Points for Credential Initialization and Administration	**pivPutData**
	pivGenerateKeyPair

6.1 Entry Points for Communication

6.1.1 pivConnect

Purpose: Connects the client-application programming interface and hence the client application itself to the PIV Card Application on a specific ICC.

Prototype:

```
status_word pivConnect(
   IN    Boolean           sharedConnection,
   INOUT sequence of bytes connectionDescription,
   OUT   handle            cardHandle
);
```

Parameters:

sharedConnection	If TRUE other client-applications can establish concurrent connections to the ICC. If FALSE and the connection is established then the calling client-application has exclusive access to the ICC.

connectionDescription	A connection description data object (tag '7F 21'). See Table 11.
	If the length of the value field of the '8x' data object in the connection description data object is zero then a list of the card readers of the type indicated by the tag of the '8x' series data object and available at the '9x' location is returned in the connectionDescription.
cardHandle	The returned opaque identifier of a communication channel to a particular ICC and hence of the card itself. cardHandle is used in all other entry points on the PIV client-application programming interface to identify which card the functionality of the entry point is to be applied.

Return Codes:
```
PIV_OK
PIV_CONNECTION_DESCRIPTION_MALFORMED
PIV_CONNECTION_FAILURE
PIV_CONNECTION_LOCKED
```

6.1.2 pivDisconnect

Purpose: Disconnect the PIV application programming interface from the PIV Card Application and the ICC containing the PIV Card Application.

Prototype:
```
status_word pivDisconnect(
   IN handle            cardHandle
);
```

Parameters:	**cardHandle**	Opaque identifier of the card to be acted upon as returned by pivConnect. The value of cardHandle is undefined upon return from pivDisconnect.

Return Codes:
```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_READER_ERROR
```

6.2 Entry Points for Data Access

6.2.1 pivSelectCardApplication

Purpose: Set the currently selected card application.

Prototype:
```
status_word pivSelectCardApplication(
   IN handle            cardHandle,
```

```
   IN sequence of byte  applicationAID,
   OUT sequence of byte applicationProperties
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
applicationAID	The AID of the card application that is to become the currently selected card application.
applicationProperties	The application properties of the selected card application. See Table 8.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_APPLICATION_NOT_FOUND
PIV_CARD_READER_ERROR
```

6.2.2 pivLogIntoCardApplication

Purpose: Establishes application security status within the PIV Card Application.

Prototype:

```
status_word pivLogIntoCardApplication(
   IN handle            cardHandle,
   IN sequence of byte  authenticators,
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
authenticators	A sequence of zero or more BER-TLV encoded authenticators to be used to authenticate the client-application to the card application and hence in establishing the initial security status in the card application context.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_AUTHENTICATOR_MALFORMED
PIV_AUTHENTICATION_FAILURE
PIV_CARD_READER_ERROR
```

6.2.3 pivGetData

Purpose: Return the entire data content of the named data object.

Prototype:

```
status_word pivGetData(
   IN handle            cardHandle,
   IN string            OID,
   OUT sequence of byte data
```

```
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
OID	Object identifier of the object whose data content is to be retrieved coded as a string; for example, "2.16.840.1.101.3.7.1.1.2.2.1"
data	Retrieved data content.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_OID
PIV_DATA_OBJECT_NOT_FOUND
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_CARD_READER_ERROR
```

6.2.4 pivLogoutOfCardApplication

Purpose: Reset the application security status of the PIV Card Application. The currently selected application after successful return from this entry point is platform-dependent.

Prototype:

```
status_word pivLogOutOfCardApplication(
   IN handle            cardHandle
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect. The cardHandle remains valid after execution of this function.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_READER_ERROR
```

6.3 Entry Points for Cryptographic Operations

6.3.1 pivCrypt

Purpose: Perform a cryptographic operation such as encryption or signing on a sequence of bytes.

Prototype:

```
status_word pivCrypt(
   IN handle              cardHandle,
   IN byte                algorithmIdentifier,
   IN byte                keyReference,
   IN sequence of byte    algorithmInput,
   OUT sequence of byte   algorithmOutput
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
algorithmIdentifier	Identifier of the cryptographic algorithm to be used for the cryptographic operation. See Table 7.
keyReference	Identifier of the on-card key to be used for the cryptographic operation. See Table 12.
algorithmInput	Sequence of bytes used as the input to the cryptographic operation.
algorithmOutput	Sequence of bytes output by the cryptographic operation.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_KEYREF_OR_ALGORITHM
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_INPUT_BYTES_MALFORMED
PIV_CARD_READER_ERROR
```

The `PIV_INPUT_BYTES_MALFORMED` error condition indicates that some property of the data to be processed such as the length or padding was inappropriate for the requested cryptographic algorithm or key.

6.4 Entry Points for Credential Initialization and Administration

6.4.1 pivPutData

Purpose: Replace the entire data content of the named data object with the provided data.

Prototype:

```
status_word pivPutData(
   IN handle            cardHandle,
   IN string            OID,
   IN sequence of byte  data
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
OID	Object identifier of the object whose data content is to be replaced coded as a string; for example, “2.16.840.1.101.3.7.1.1.2.2.1”
data	Data to be used to replace in its entirety the data content of the named data object.

Return Codes:
```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_OID
PIV_DATA_OBJECT_NOT_FOUND
PIV_INSUFFICIENT_CARD_RESOURCE
PIV_CARD_READER_ERROR
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
```

6.4.2 pivGenerateKeyPair

Purpose: Generates an asymmetric key pair in the currently selected application.

If the provided key reference exists and the cryptographic mechanism associated with the reference data identified by this key reference is the same as the provided cryptographic mechanism, then the generated key pair replaces in entirety the key pair currently associated with the key reference.

Prototype:
```
status_word pivGenerateKeyPair(
   IN handle               cardHandle,
   IN byte                 keyReference,
   IN byte                 cryptographicMechanism,
   OUT sequence of byte    publicKey
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
keyReference	The key reference of the generated key pair.
cryptographicMechanism	The type of key pair to be generated. See Table 20.
publicKey	BER-TLV data objects defining the public key of the generated key pair. See Table 21.

Return Codes:
```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_INVALID_KEY_OR_KEYALG_COMBINATION
PIV_UNSUPPORTED_CRYPTOGRAPHIC_MECHANISM
PIV_CARD_READER_ERROR
```

7. Part 3: End-Point PIV Card Application Card Command Interface

The Table 15 lists the card commands surfaced by the PIV Card Application at the card edge of the ICC containing it. All PIV Card Application card commands shall be supported by a PIV Card Application. Card commands indicated with a 'Yes' in the Command Chaining column shall support command chaining for transmitting a data string too long for a single command as defined in ISO/IEC 7816-4 [1].

Table 15. PIV Card Application Card Commands

Type	Name	Contact Interface	Contactless Interface	Security Condition for Use	Command Chaining
PIV Card Application Card Commands for Data Access	**SELECT**	Yes	Yes	Always	No
	GET DATA	Yes	Yes	Data Dependent. See Table 1.	No
PIV Card Application Card Commands for Authentication	**VERIFY**	Yes	No	Always	No
	CHANGE REFERENCE DATA	Yes	No	Application PIN	No
	RESET RETRY COUNTER	Yes	No	PIN Unblocking Key	No
	GENERAL AUTHENTICATE	Yes	Yes (See Note)	Key Dependent.	Yes
PIV Card Application Card Commands for Credential Initialization and Administration	**PUT DATA**	Yes	No	PIV Card Application Administrator	Yes
	GENERATE ASYMMETRIC KEY PAIR	Yes	No	PIV Card Application Administrator	Yes

The PIV Card Application shall return the status word of ‘6A81’ (Function not supported) when it receives a card command on the contactless interface marked “No” in the Contactless Interface column in Table 15.

Note: Cryptographic protocols using asymmetric keys that require PIN shall not be used on the contactless interface.

7.1 PIV Card Application Card Commands for Data Access

7.1.1 SELECT Card Command

The SELECT card command sets the currently selected application. The PIV Card Application shall be selected by providing its application identifier

'A0 00 00 03 08 00 00 10 00 vv vv’

in the data field of the SELECT command where 'vv vv' is the version of the PIV Card Application to be made the currently selected application. The AID of the initial version of the PIV Card Application is

'A0 00 00 03 08 00 00 10 00 01 00'

There shall be at most one PIV Card Application on any ICC. The PIV Card Application can also be made the currently selected application by providing the right-truncated version; that is, without the two-byte version number, 'vv vv'; in the data field of the SELECT command

'A0 00 00 03 08 00 00 10 00'

The complete AID, including the two-byte version, of the PIV Card Application that became the currently selected application upon successful execution of the SELECT command shall be returned in the application property template.

If the currently selected application is the PIV Card Application when the SELECT APPLICATION command is given and the AID in the data field of the SELECT APPLICATION is either the AID of the PIV Card Application or the right-truncated version thereof, then the PIV Card Application shall continue to be the currently selected application and the setting of all security status indicators in the PIV Card Application shall be unchanged.

If the currently selected application is the PIV Card Application when the SELECT APPLICATION command is given and the AID in the data field of the SELECT APPLICATION is neither the AID of the PIV Card Application or the right-truncated version thereof, then the PIV Card Application shall be deselected and all PIV Card Application security status indicators shall be set to FALSE.

Command Syntax

CLA	'00'
INS	'A4'
P1	'04'
P2	'00'
L_c	Length of application identifier
Data Field	Application identifier of the PIV Card Application, possibly right-truncated
L_e	Length of application property template

Response Syntax

Data Field	Application property template
SW1-SW2	Status word

SW1	SW2	Meaning
'6A'	'82'	Application not found
'90'	'00'	Successful execution

7.1.2 GET DATA Card Command

The GET DATA card command retrieves the data content of the single data object whose tag is given in the data field.

Command Syntax

CLA	'00'
INS	'CB'
P1	'3F'
P2	'FF'
L_c	'05'
Data Field	See Table 16.
L_e	Number of data content bytes to be retrieved.

Table 16. Data Objects in the Data Field of the GET DATA Card Command

Name	Tag	M/O	Comment
Tag list	'5C'	M	BER-TLV tag of the data object to be retrieved. See Table 6.

Response Syntax

Data Field	BER-TLV with the tag '53' containing in the value field the requested data object.
SW1-SW2	Status word

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available
'69'	'82'	Security status not satisfied
'6A'	'82'	Data object not found
'90'	'00'	Successful execution

7.2 PIV Card Application Card Commands for Authentication

7.2.1 VERIFY Card Command

The VERIFY card command initiates the comparison in the card of the reference data indicated by the key reference with authentication data in the data field of the command.

Only key references specific to the PIV Card Application; i.e. local key references, shall be verified by the PIV Card Application VERIFY command.

If the current value of the retry counter associated with the key reference is zero, then the comparison shall not be made and the PIV Card Application shall return the status word '69 83'.

If the reference data in the command data field does not satisfy the criteria in Section 3.5.3, the PIV Card Application shall return the status word '6A 80'.

If the card command succeeds, then the security status of the key reference shall be set to TRUE and the retry counter associated with the key reference shall be set to the reset retry value associated with the key reference.

If the card command fails, then the security status of the key reference shall be set to FALSE and the retry counter associated with the key reference shall be decremented by one.

The initial value of the retry counter and reset retry value associated with the key reference; i.e. the number of successive failures (retries) before the retry counter associated with the key reference reaches zero, is issuer dependent.

Command Syntax

CLA	'00'
INS	'20'
P1	'00'
P2	Key reference. See Table 12.
L_c	‘08’
Data Field	PIN reference data as described in 3.5.3
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'CX'	Verification failed, X indicates the number of further allowed retries
‘69’	‘83’	Authentication method blocked
‘6A’	‘80’	Incorrect parameter in command data field
'6A'	'88'	Key reference not found
'90'	'00'	Successful execution

7.2.2 CHANGE REFERENCE DATA Card Command

The CHANGE REFERENCE DATA card command initiates the comparison of the verification data with the current value of the reference data and if this comparison is successful replaces the reference data with new reference data. Only reference data associated with key references specific to the PIV Card Application can be changed by this PIV Card Application command.

Only reference data associated with key references specific to the PIV Card Application; i.e. local key references, shall be changed by the PIV Card Application CHANGE REFERENCE DATA command.

If the current value of the retry counter associated with the key reference is zero, then the reference data associated with the key reference shall not be changed and the PIV Card Application shall the status word '69 83'.

If the card command succeeds, then the security status of the key reference shall be set to TRUE and the retry counter associated with the key reference shall be set to the reset retry value associated with the key reference.

If the card command fails, then the security status of the key reference shall be set to FALSE and the retry counter associated with the key reference shall be decremented by one.

The initial value of the retry counter and the reset retry value associated with the key reference; i.e. the number of successive failures (retries) before the retry counter associated with the key reference reaches zero, is issuer dependent.

If the either the current reference data or the new reference data in the command field of the command does not satisfy the criteria in Section 3.5.3, the PIV Card Application shall not change the reference data associated with the key reference and shall return the status word '6A 80'.

Command Syntax

CLA	'00'
INS	'24'
P1	'00'
P2	Key reference. See Table 12.
L_c	‘10’
Data Field	Current PIN reference data concatenated without delimitation with the new PIN reference data, both PINs as described in 3.5.3
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'CX'	Verification failed, X indicates the number of further allowed retries or resets
‘69’	‘83’	Authentication method blocked
‘6A’	‘80’	Incorrect parameter in command data field
'6A'	'88'	Key reference not found
'90'	'00'	Successful execution

7.2.3 RESET RETRY COUNTER Card Command

The RESET RETRY COUNTER card command resets the retry counter of the key reference to its initial value and changes the reference data associated with the key reference. The command enables recovery of the PIN card application in the case that the cardholder has forgotten a PIV Card Application PIN.

Only retry counters associated with key references specific to the PIV Card Application; i.e. local key references, shall be reset by the PIV Card Application RESET RETRY COUNTER command.

If the current value of the reset counter associated with the key reference is zero, then retry counter associated with the key reference shall not be reset and the PIV Card Application shall the status word '69 83'.

If the card command succeeds, then the retry counter associated with the key reference shall be set to the reset retry value associated with the key reference. Neither the security status of the key reference or the reset counter shall be changed.

If the card command fails, then the security status of the key reference shall be set to FALSE and the reset counter associated with the key reference shall be decremented by one.

The initial reset counter associated with the key reference; i.e. the number of failures of the RESET RETRY COUNTER command before the reset counter associated with the key reference reaches zero, is issuer dependent.

If the either the reset retry counter reference data (PUK) or the new reference data (PIN) in the command field of the command does not satisfy the criteria in Section 3.5.3, the PIV Card Application shall not reset the retry counter associated with the key reference and shall return the status word '6A 80'.

Command Syntax

CLA	'00'
INS	'2C'
P1	'00'
P2	Key reference. See Table 12.
L_c	'10'
Data Field	Reset retry counter reference data (PUK) concatenated without delimitation with the new reference data (PIN), both PUK and PIN as described in 3.5.3
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'CX'	Verification failed, X indicates the number of further allowed retries
'69'	'83'	Authentication method blocked
'6A'	'80'	Incorrect parameter in command data field
'6A'	'88'	Key reference not found
'90'	'00'	Successful execution

7.2.4 GENERAL AUTHENTICATE Card Command

The GENERAL AUTHENTICATE card command performs a cryptographic operation such as an authentication protocol using the data provided in the data field of the command and returns the result of the cryptographic operation in the response data field.

The GENERAL AUTHENTICATE command shall be used to authenticate the card or a card application to the client-application (INTERNAL AUTHENTICATE), to authenticate an entity to the card

(EXTERNAL AUTHENTICATE), and to perform a mutual authentication between the card and an entity external to the card (MUTUAL AUTHENTICATE).

The GENERAL AUTHENTICATE command shall be used to realize the signing functionality on the PIV client-application programming interface. Data sent to the card is expected to be hashed off-card.

The GENERAL AUTHENTICATE command supports command chaining to permit the uninterrupted transmission of long command data fields to the PIV Card Application. If a card command other than the GENERAL AUTHENTICATICATE command is received by the PIV Card Application before the termination of a GENERAL AUTHENTICATE chain, the PIV Card Application shall rollback to the state it was in immediately prior to the reception of the first command in the interrupted chain. In other words, an interrupted GENERAL AUTHENTICATE chain has no effect on the PIV Card Application.

Command Syntax

CLA	'00' or '10' indicating command chaining.
INS	'87'
P1	Algorithm reference
P2	Key reference
L_c	Length of data field
Data Field	See Table 17.
L_e	Absent or length of expected response

Table 17. Data Objects in the Dynamic Authentication Template (Tag '7C')

Name	Tag	M/O	Description
Witness	'80'	C	Demonstration of knowledge of a fact without revealing the fact. An empty witness is a request for a witness.
Challenge	'81'	C	One or more random numbers or byte sequences to be used in the authentication protocol.
Response	'82'	C	A sequence of bytes encoding a response step in an authentication protocol.
Committed challenge	'83'	C	Hash-code of a large random number including one or more challenges
Authentication code	'84'	C	Hash-code of one or more data fields and a witness data object.

The data objects that appear in the dynamic authentication template (tag '7C') in the data field of the GENERAL AUTHENTICATE card command depend on the authentication protocol being executed.

Response Syntax

Data Field	Absent or authentication-related data
SW1-SW2	Status word

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available
’69’	‘82’	Security status not satisfied
‘6A’	‘80’	Incorrect parameter in command data field
'6A'	'86'	Incorrect parameter in P1 or P2
'90'	'00'	Successful execution

7.3 PIV Card Application Card Commands for Credential Initialization and Administration

7.3.1 PUT DATA Card Command

The PUT DATA card command completely replaces the data content of a single data object in the PIV Card Application with new content.

Command Syntax

CLA	'00' or '10' indicating command chaining.
INS	'DB'
P1	‘3F'
P2	‘FF’
L_c	Length of data field
Data Field	See Table 18.
L_e	Empty

Table 18. Data Objects in the Data Field of the PUT DATA Card Command

Name	Tag	M/O	Description
Tag list	'5C'	M	Tag of the data object whose data content is to be replaced. See Table 6.
Data	‘53’	M	Data with tag ‘53’ as an unstructured byte sequence.

Response Syntax

Data Field	Absent or authentication-related data
SW1-SW2	Status word

SW1	SW2	Meaning
'69'	'82'	Security status not satisfied
'6A'	'84'	Not enough memory
'90'	'00'	Successful execution

7.3.2 GENERATE ASYMMETRIC KEY PAIR Card Command

The GENERATE ASYMMETRIC KEY PAIR card command initiates the generation and storing in the card of the reference data of an asymmetric key pair, i.e., a public key and a private key. The public key of the generated key pair is returned as the response to the command. If there is reference data currently associated with the key reference, it is replaced in full by the generated data.

Command Syntax

CLA	'00' or '10' indicating command chaining.
INS	'47'
P1	'00'
P2	Non-zero key reference to be assigned to the generated asymmetric key pair.
L_c	Length of data field
Data Field	Control reference template. See Table 19.
L_e	Length of public key of data object template

Table 19. Data Objects in the Template (Tag 'AC')

Name	Tag	M/O	Description
Cryptographic mechanism identifier	'80'	M	See Table 20.
Parameter	'81'	C	Specific to the cryptographic mechanism

Table 20. Cryptographic Mechanism Identifiers

Cryptographic Mechanism Identifier	Description	M/O	Parameter
'00'-'04'	RFU		
'06'	RSA 1024	M	Optional public exponent encoded big-endian
'07'	RSA 2048	O	Optional public exponent encoded big-endian
'05'	RSA 3072	O	Optional public exponent encoded big-endian
'08'-'0D'	RFU		
'0E'	ECC: Curve P-224	O	None
'0F'	ECC: Curve K-233	O	None
'10'	ECC: Curve B-233	O	None
'11'	ECC: Curve P-256	O	None
'12'	ECC: Curve K-283	O	None
'13'	ECC: Curve P-283	O	None

All other cryptographic mechanism identifier values are reserved for future use.

Response Syntax

Data Field	Data objects of public key of generated key pair. See Table 21.
SW1-SW2	Status word

Table 21. Data Objects in the Template (Tag '7F49')

Name	Tag
Public key data objects for RSA	
Modulus	'81'
Public exponent	'82'
Public key data objects for ECDSA	
Prime	'81'
First coefficient	'82'
Second coefficient	'83'
Generator	'84'
Order	'85'
Point	'86'

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available
'69'	'82'	Security status not satisfied
'6A'	'80'	Incorrect parameter in command data field; e.g. unrecognized cryptographic mechanism
'6A'	'86'	Incorrect parameter P2; cryptographic mechanism of reference data to be generated different than cryptographic mechanism of reference data of given key reference
'90'	'00'	Successful execution

Appendix A—PIV Data Model

The PIV data model number is 0x10, and the data model version number is 0x01.

The SP800-73 Part 3 specification does not provide mechanisms to read partial contents of a PIV container. Individual access to the TLV elements within a container is not supported. Part 3 compliant cards shall return all the TLV elements of a container in the physical order listed for that container in this data model.
Both single-chip/dual-interface and dual-chip implementations shall be feasible. In the single-chip/dual-interface configuration, the PIV Card Application shall be provided the information regarding which interface is in use. In the dual-chip configuration, a separate PIV Card Application shall be loaded on each chip.

Buffer Description	Container ID	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	0x0101	1651	PIN	Contact	M
Card Holder Fingerprints	0x6010	7768	PIN	Contact	M
Printed Information	0x3001	106	PIN	Contact	O
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

Note that all data elements in the following tables are mandatory unless specified as optional.

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
Capability Tuples (CTs)	0xFB	Fixed	0
Status Tuples (STs)	0xFC	Fixed	0

Card Capabilities Container	0xDB00	Always Read	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Next CCC	0xFD	Fixed	0
Extended Application CardURL (optional)	0xE3	Fixed	48
Security Object Buffer (optional)	0xB4	Fixed	48
Error Detection Code	0xFE	LRC	0

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

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

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

Printed Information	0x3001	PIN	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Name	0x01	Fixed Text	32
Employee Affiliation line 1	0x02	Fixed Text	20
Employee Affiliation line 2	0x03	Fixed Text	20
Expiration date	0x04	Fixed Text	9
Agency Card Serial Number	0x05	Fixed Text	10
Issuer Identification	0x06	Fixed Text	15
Error Detection Code	0xFE	LRC	0

Card Holder Facial Image	0x6030	PIN	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Image for Visual Verification	0xBC	Variable	12704
Error Detection Code	0xFE	LRC	0

X.509 Certificate for Digital Signature	0x0100	pkiCompute -PIN Always	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Certificate	0x70	Variable	1856
CertiInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

X.509 Certificate for Key Management	0x0102	pkiCompute – PIN	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Certificate	0x70	Variable	1856
CertiInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

X.509 Certificate for Card Authentication	0x0500	Asymmetric – pkiCompute - Always Symmetric – See CCC / CHUID	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Certificate	0x70	Variable	1856
CertiInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
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	0xBB	Variable	900
Error Detection Code	0xFE	LRC	0

The CertInfo byte in certificates identified above shall be encoded as follows:

```
CertInfo::= BIT STRING {
                    CompressionTypeMsb(0),      // 0 = no compression and 1 = gzip compression.
                    CompressionTypeLsb(1),      // shall be set to ‘0’ for PIV Applications
                    IsX509(2),                  // shall be set to ‘0’ for PIV Applications
                    RFU3(3),
                    RFU4(4),
                    RFU5(5),
                    RFU6(6),
                    RFU7(7)
                    }
```

Appendix B—Examples of the Use of GENERAL AUTHENTICATE

B.1 Authentication of the PIV Card Application Administrator

The PIV Card Application Administrator is authenticated by the PIV Card Application using a challenge/response protocol. A challenge retrieved from the PIV Card Application is encrypted by the client-application and returned to the PIV Card Application associated with key reference '9B', the key reference to the PIV Card Application Administration Key. The PIV Card Application decrypts the response using this reference data and the algorithm associated with the key reference; that is 3 Key Triple DES – ECB (algorithm identifier '00'). If this decrypted value matches the previously provided challenge, then the security status indicator of the PIV Card Application Administrator is set to TRUE within the PIV Card Application.

Table 22 shows the GENERAL AUTHENTICATE card commands sent to the PIV Card Application to realize this particular challenge/response protocol.

Table 22. Authentication of PIV Card Application Administrator

Command	Response	Comment
'00 87 00 00 04 7C 02 81 00'		Client-application requests a challenge from the PIV Card Application
	'7C 0A 81 08 01 02 03 04 05 06 07 08'	Challenge returned to client-application by the PIV Card Application
'00 87 00 9B 0C 7C 0A 82 08 88 77 66 55 44 33 22 11'		Client-application returns the encryption of the challenge ('88 77 66 55 44 33 22 11') referencing algorithm '00' and key reference '9B'. See Tables 7 and 12.
	'9000'	PIV Card Application indicates successful authentication of PIV Card Application Administrator after decrypting '88 77 66 55 44 33 22 11' using the referenced algorithm and key and getting '01 02 03 04 05 06 07 08'

B.2 Validation of the PIV Card Application

The PIV Card Application is validated by first retrieving the X.509 Certificate of the PIV Authentication Key (OID 2.16.840.1.101.3.7.2.1.1) and verifying the signature on this certificate. Assuming the certificate is valid and current, the client-application requests the PIV Card Application to encrypt a challenge using the private key associated with this certificate; i.e. key reference '9A', algorithm identifier '06'. The response is decrypted using the public key in the certificate. If the decrypted response matches the challenge, then the PIV Card Application is validated.

Table 23 shows the GENERAL AUTHENTICATE card commands sent to the PIV Card Application to realize the validation of the PIV Card Application.

Table 23. Validation of the PIV Card Application Using GENERAL AUTHENTICATE

Command	Response	Comment
’00 87 06 9A 0E 7C 0C 82 00 81 08 01 02 03 04 05 06 07 08’		Client-application sends a challenge to the PIV Card Application indicating the reference data associated with key reference ‘9A’ is to be used with algorithm ‘06’. See Tables 7 and 12.
	‘7C 0A 82 08 88 77 66 55 44 33 22 11’	PIV Card Application returns the encryption of the challenge (’88 77 66 55 44 33 22 11’) using the indicated key reference data and algorithm.

The same use of GENERAL AUTHENTICATE can be used to achieve a signing of a byte sequence such as a hash by the PIV Card Application. One need only indicate which algorithm and which key are to be used by setting values of the P1 and P2 parameters respectively.

Note that for exposition purposes this example uses only a 8-byte challenge and response with a 1024-bit RSA key. In actual usage a challenge and response more appropriate for this cryptographic algorithm would be used.

Appendix C—PIV Authentication Use Cases

To provide guidance on the usage and behavior supported by the PIV Card, PIV authentication use cases and application scenarios are described in this section. FIPS 201 describes PIV authentication as the "process of establishing confidence in the identity of the cardholder presenting a PIV Card." The fundamental goal of using the PIV Card is to authenticate the identity of the cardholder to a system or person that is controlling access to a protected resource or facility. This end goal may be reached by various combinations of one or more of the validation steps described below:

+ Card Validation (CardV) — This is the process of verifying that a PIV Card is authentic (i.e., not a counterfeit card) and has not been subjected to tampering or alteration. Card validation mechanisms include:

    - Visual inspection of the tamper-proofing and tamper-resistant features of the PIV Card as per Section 4.1.2 of FIPS 201,

    - Use of cryptographic challenge-response schemes with symmetric keys,

    - Use of asymmetric authentication schemes to validate private keys embedded within the PIV Card.

+ Credential Validation (CredV) — This is the process of verifying the various types of credentials (such as visual credentials, CHUID, biometrics, PIV keys and certificates) held by the PIV Card. Credential validation mechanisms include:

    - Visual inspection of PIV Card visual elements (such as the photo, the printed name, and rank, if present),

    - Verification of certificates on the PIV Card,

    - Verification of signatures on the PIV biometrics and the CHUID,

    - Checking the expiration date,

    - Checking the revocation status of the credentials on the PIV Card.

+ Cardholder Validation (HolderV) — This is the process of establishing that the PIV Card is in the possession of the individual who is the legitimate owner of the card. Classically, identity authentication is achieved using one or more of these factors: a) something you have, b) something you know, and c) something you are. The assurance of the authentication process increases with the number of factors used. In the case of the PIV Card, these three factors translate as follows: a) something you have – possession of a PIV Card, b) something you know – knowledge of the PIN, and c) something you are – the visual characteristics of the cardholder, and the live fingerprint samples provided by the cardholder. Thus, mechanisms for PIV cardholder validation include:

    - Presentation of a PIV Card by the cardholder,

    - Matching the visual characteristics of the cardholder with the photo on the PIV Card,

- Matching the PIN provided with the PIN on the PIV Card,
- Matching the live fingerprint samples provided by the cardholder, with the biometric information embedded within the PIV Card.

C.1 Use Case Diagrams

This section describes the activities and interactions involved in interoperable usage and authentication of the PIV Card. The use cases represent how a relying party will authenticate the cardholder (regardless of which agency issued the card) in order to provide access to its systems or facilities. These activities and interactions are represented in functional use case diagrams. These diagrams are not intended to provide syntactical commands or API function names.

Each of the PIV authentication mechanisms described in this section can be broken into a sequence of one or more validation steps where Card, Credential, and Cardholder validation is performed. In the use case illustrations, the validation steps are marked as CardV, CredV and HolderV to signify Card, Credential and Cardholder validation respectively.

Depending upon the assurance provided by the actual sequence of validation steps in a given PIV authentication mechanism, relying parties can make appropriate decisions for granting access to protected resources based on a risk analysis.

C.1.1 Authentication using PIV Visual Credentials

This is the use case where a human guard authenticates the cardholder using the visual credentials held by the PIV Card, and is illustrated in Figure C-1.

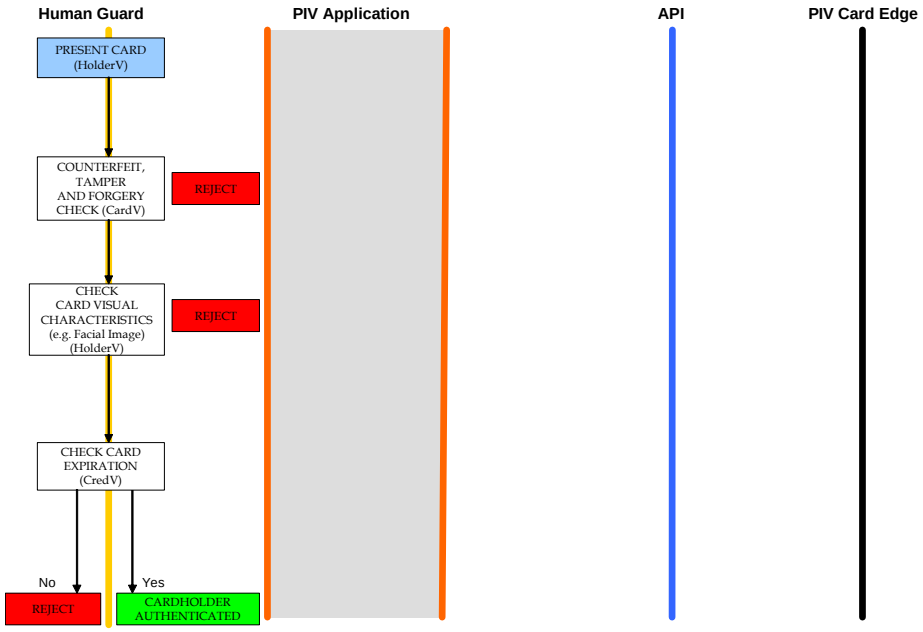

Figure C-1: Authentication using PIV Visual Credentials

C.1.2 Authentication using PIV CHUID

The PIV CHUID may be used for authentication in several variations. The use of the PIV Card to implement a PACS Low assurance profile is illustrated in Figure C-2. The minimum set of authentication data that must be transmitted from the PIV Application to the Local System is application dependent and therefore not defined in this Specification.

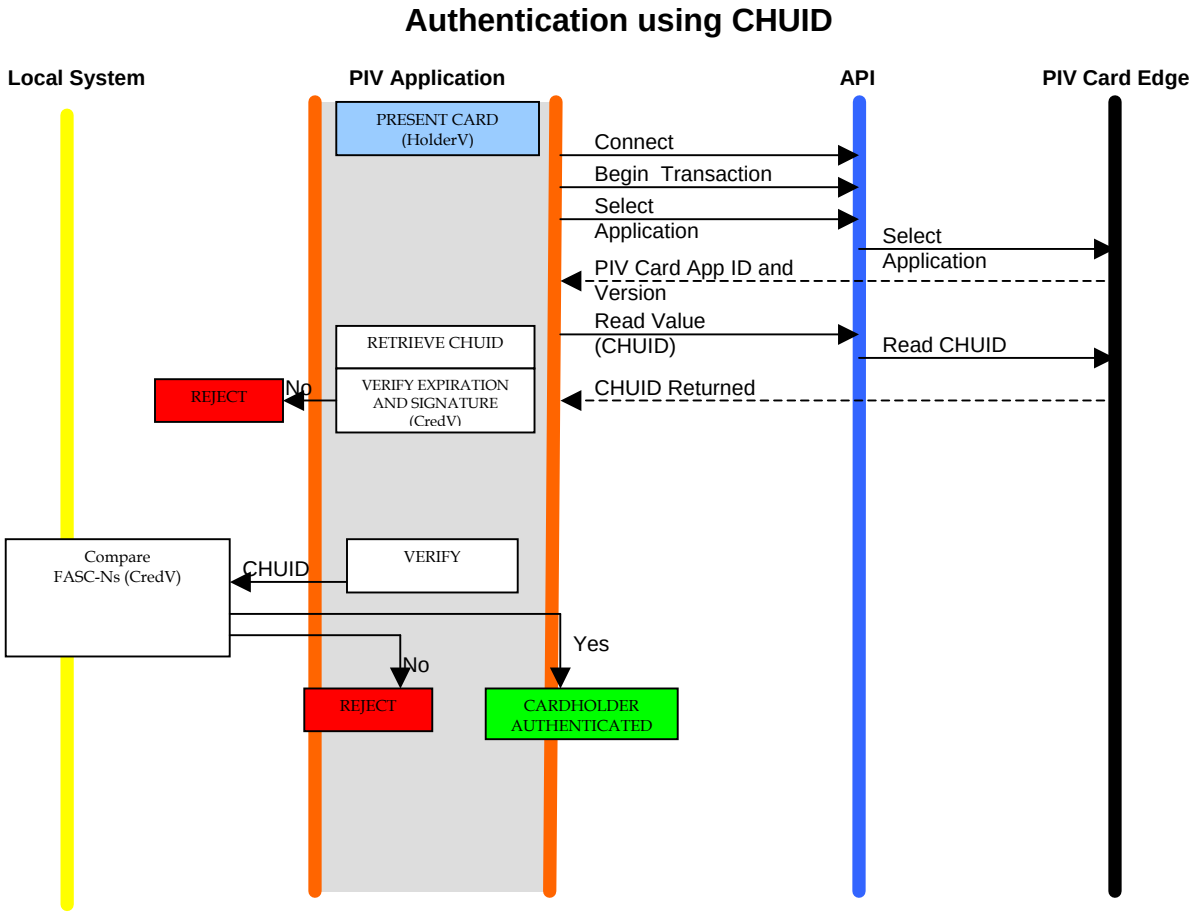

Figure C-2: Authentication using PIV CHUID

C.1.3 Authentication using PIV Biometrics

The general use case for authentication using the PIV biometric is illustrated in Figure C-3.

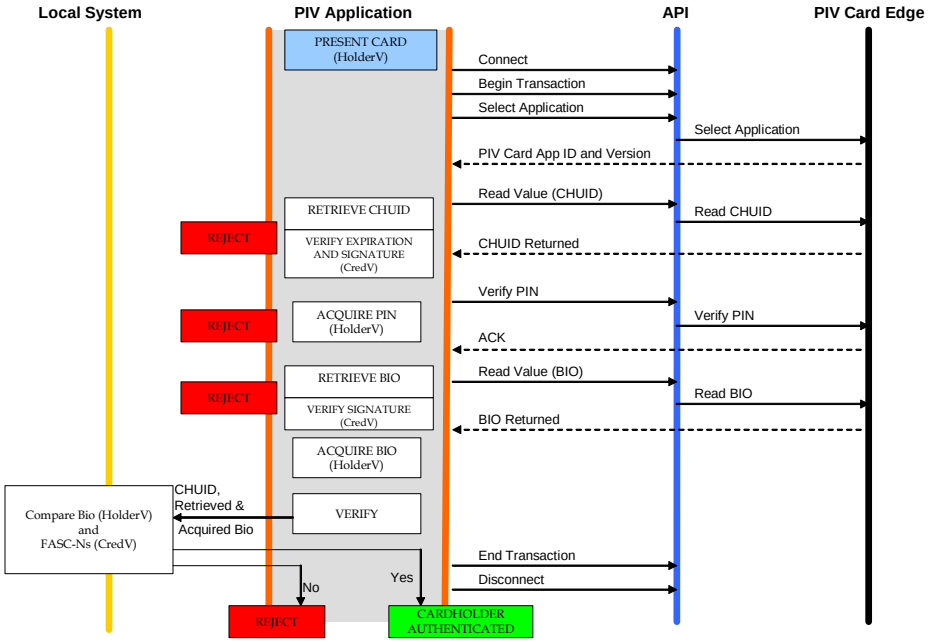

Figure C-3: Authentication using PIV Biometrics

The assurance of authentication using the PIV biometric can be further increased if the live biometric sample is collected in an attended environment, with a human overseeing the process. This use case is illustrated in Figure C-4.

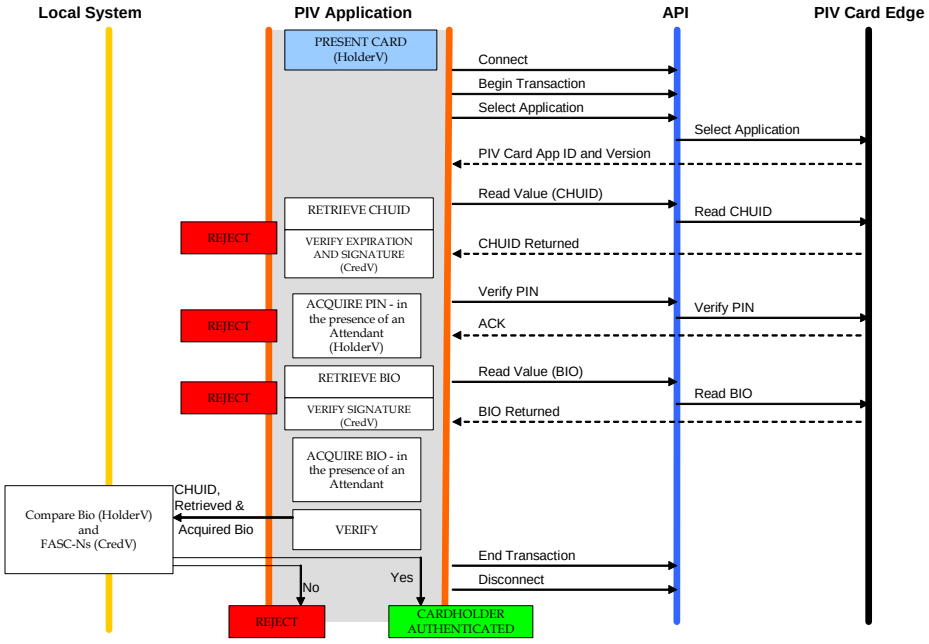

Figure C-4: Authentication using PIV Biometrics (Attended)

C.1.4 Authentication using PIV Authentication Key

The use case for authentication using the PIV Authentication Key is illustrated in Figure C-5.

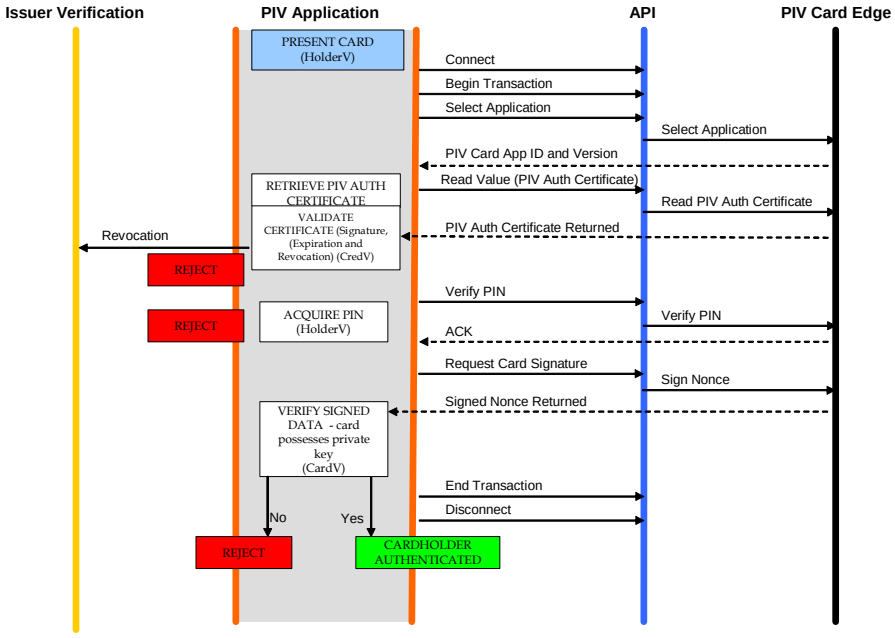

Figure C-5: Authentication using PIV Authentication Key

C.2 Summary Table

The following table summarizes the types of validation activities that are included in each of the PIV authentication mechanisms described earlier in this section.

PIV Authentication Mechanism	Card Validation Steps (CardV)	Credential Validation Steps (CredV)	Cardholder Validation Steps (HolderV)
PIV Visual Authentication	1. Counterfeit, tamper and forgery check	1. Expiration check	Possession of Card Match of card visual characteristics with cardholder
PIV CHUID		Expiration check CHUID signature check (optional)	Possession of Card
PIV Biometric (Unattended)		Expiration check CHUID signature check (optional) PIV Bio signature check (optional)	Possession of Card Match PIN Match holder's bio with PIV bio
PIV Biometric (Attended)		Expiration check CHUID signature check (optional) PIV Bio signature check (optional)	Possession of Card Match PIN Match of holder's bio to PIV bio *in view of attendant*

PIV Authentication Mechanism	Card Validation Steps (CardV)	Credential Validation Steps (CredV)	Cardholder Validation Steps (HolderV)
PIV Authentication Key	1. Perform challenge response with a PIV asymmetric key, and validate signature on response	Card expiration check Certificate validation of a PIV certificate	Possession of Card Match PIN provided by holder with PIV PIN

Appendix D—Terms, Acronyms, and Notation

D.1 Terms

Application Identifier	A globally unique identifier of a card application as defined in ISO/IEC 7816-4.
Application Session	The period of time within a card session between when a card application is selected and a different card application is selected or the card session ends.
Authenticatable Entity	An entity that can successfully participate in an authentication protocol with a card application.
BER-TLV Data Object	A data object coded according to ISO/IEC 8825-2.
Card	An integrated circuit card.
Card Application	A set of data objects and card commands that can be selected using an application identifier.
Card Interface Device	An electronic device that connects an integrated circuit card and the card applications therein to a client application.
Card Reader	Synonym for card interface device.
Client Application	A computer program running on a computer in communication with a card interface device.
Data Object	An item of information seen at the card command interface for which are specified a name, a description of logical content, a format and a coding.
Interface Device	Synonym for card interface device.
Key Reference	A 6-bit identifier of cryptographic material used in a cryptographic protocol such as an authentication or a signing protocol.
MSCUID	An optional legacy identifier included for compatibility with Common Access Card and Government Smart Card Interoperability Specifications.
Object Identifier	A globally unique identifier of a data object as defined in ISO/IEC 8824-2.
Reference Data	Cryptographic material used in the performance a cryptographic protocol such as an authentication or a signing protocol.
Status Word	Two bytes returned by an integrated circuit card after processing any command that signify the success of or errors encountered during said processing.
Template	A (constructed) BER-TLV data object whose value field contains specific BER-TLV data objects.

D.2 Acronyms

AES	Advanced Encryption Standard
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASN.1	Abstract Syntax Notation
BER	Basic Encoding Rules
BSI	Basic Services Interface
CBC	Circular Binary Coding
CBEFF	Common Biometric Exchange Formats Framework
CCC	Card Capability Container
CLA	Class (first) byte of a card command
CHUID	Card Holder Unique IDentifier
DES	Data Encryption Standard
DNS	Domain Name Server
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EMR	Electro Magnetic Radiation
FASC-N	Federal Agency Smart Credential Number
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GSC-IAB	Government Smart Card Interagency Advisory Board
GSC-IS	Government Smart Card Interoperability Specification
GUID	Global Unique Identification Number

ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
INS	Instruction (second) byte of a card command
IP	Internet Protocol
IR	Infra Red
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
LSB	Least Significant Bit
MRTD	Machine Readable Travel Document
MSB	Most Significant Bit
OID	Object Identifier
OMB	Office of Management and Budget
P1	First parameter of a card command
P2	Second parameter of a card command
PACS	Physical Access Control System
PC/SC	Personal Computer/Smart Card
PIN	Personal Identification Number
PIV	Personal Identity Verification
PIX	Proprietary Identifier eXtension
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PUK	PIN Unblocking Key
RFU	Reserved for Future Use
RID	Registered application provider IDentifier
RSA	Rivest, Shamir, Aldeman
SCEPACS	Smart Card Enabled Physical Access Control System

SCP	ETSI Smart Card Project
SP	Special Publication
SW1	First byte of a two-byte status word
SW2	Second byte of a two-byte status word
TIG	Technical Implementation Guidance
TLV	Tag-Length-Value
URL	Uniform Resource Locator
VM	Virtual Machine

D.3 Notation

The sixteen hexadecimal digits shall be denoted using the alphanumeric characters 0, 1, 2…, A, B, C, D, E, and F. A byte consists of two hexadecimal digits, for example, ‘2D’. A sequence of bytes may be enclosed in single quotation marks, for example 'A0 00 00 01 16' rather than given as a sequence of individual bytes, 'A0’ ‘00’ ‘00’ ‘01’ ‘16'.

A byte can also be represented by bits b8 to b1, where b8 is the most significant bit (MSB) and b1 is the least significant bit (LSB) of the byte. In textual or graphic representations, the leftmost bit is the MSB. Thus, for example, the most significant bit, b8, of ‘80’ is 1 and the least significant bit, b1, is 0.

All bytes specified as RFU shall be set to '00' and all bits specified as reserved for future use shall be set to 0.

All lengths shall be measured in number of bytes unless otherwise noted.

Data objects in templates are described as being mandatory (M), optional (O) or conditional (C). 'Mandatory' means the data object shall appear in the template. 'Optional' means the data object may appear in the template. In the case of conditional data objects, the conditions under which they are required are provided in a footnote to the table.

In other tables the M/O column identifies properties of the PIV Card Application that shall be present (M) or may be present (O).

BER-TLV data object tags are represented as byte sequences as described above. Thus, for example, ‘4F’ is the interindustry data object tag for an application identifier and ‘7F 60’ is the interindustry data object tag for the biometric information template.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this standard are to be interpreted as described in IETF RFC 2119, Key Words for Use in RFCs to Indicate Requirement Levels [6].

Appendix E—References

[1] ISO/IEC 7816 (Parts 4, 5, 6, 8, and 9), *Information technology — Identification cards — Integrated circuit(s) cards with contacts.*

[2] ISO/IEC 8824-2:2002, *Information technology -- Abstract Syntax Notation One (ASN.1): Information object specification.*

[3] 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).*

[4] NIST Federal Information Processing Standards Publication 201, *Personal Identity Verification for Federal Employees and Contractors*, February, 2005.

[5] PACS v2.2, *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.

[6] IETF RFC 2119, "Key Words for Use in RFCs to Indicate Requirement Levels," March, 1997.

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

[8] 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.