NIST Special Publication 800-73
2nd Draft

Interfaces for Personal Identity Verification

National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

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 08, 2005

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

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
HratchG. Semerjian, Acting Director

NOTE FOR REVIEWERS

In response to the comments received on Draft SP 800-73, NIST has revised the SP 800-73 to include an optional migration path for agencies currently implementing GSC-IS v2.1. High level changes from the January 31, 2005 version of the SP 800-73 are the following:

1. PIV object discovery through a common name space is defined in accordance with NISTIR 6887. See Section 2.1.2 of this document for specific information.

2. Registered Identifier (RID) for PIV applications beyond the mandatory Card Capability Container (CCC) can be specific to the agency. The Extended Application CardURL may be used to point to applications beyond PIV.

3. Reconciled tag value conflicts in GSC-IS v2.1 for asymmetric keys data elements between Common Access Card (CAC) and GSC data model.

4. CHUID tag value for expiration date is modified from 0x40 to 0x35. The GUID shall be present. If assigned, the GUID should be encoded as IPv6 address. Otherwise, its value should be 0x00. The FASC-N option has been modified in accordance with PACS v2.2 to concatenate system code and credential number to enable 9,999,999,999 credentials per issuer.

5. The PIV data model includes the mandatory and optional data containers specified in FIPS 201. The PIV data model also includes CCC, Printed Information, and Security Object data containers which are beyond FIPS 201 specification.

6. Section 2 of this document provides an optional specification to support applications that currently implement GSC-IS v2.1. The remainder of this document provides the mandatory specification to support PIV interoperability objectives.

Please submit your SP 800-73 comments using the comment template form provided on the http://www.csrc.nist.gov/piv-project/fips201-support-docs.html website. Please include the submitter's name and organization in the header section of the spreadsheet. This will greatly facilitate processing of comments by NIST.

Comments should be submitted to DraftFips201@nist.gov. It is requested that Federal organizations submit one consolidated/coordinated set of comments. Also, include “Comments on Public Draft SP 800-73” in the subject line. The comment period closes at 5:00 EST (US and Canada) on March 22, 2005.

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, 73 pages (March 08, 2005)

Acknowledgements

The authors 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. 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. Introduction

The Homeland Security Presidential Directive HSPD-12 called for new standards 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) was developed to establish standards for identity credentials. This document, Special Publication 800-73 (SP 800-73), specifies interface requirements for the 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. 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 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 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 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 Appendix B, 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 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 is organized as follows:

+ Section 1, provides a scope, purpose, migration strategy for agencies that have implemented GSC-IS v2.1, and the PIV data model.

+ Section 2, provides a subset of GSC-IS v2.1 specification as an optional reference that is refined to support the PIV application during agency transitions from GSC-IS v2.1 to the mandatory specifications in this standard.

+ 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.

+ Appendix A, Provides the PIV Data Model.

+ Appendix B, Use of GENERAL AUTHENTICATE, shows how the GENERAL AUTHENTICATE card command is use to perform various required cryptographic protocols including authentication using a symmetric PIV card application key and signing using an asymmetric PIV card application key.

+ Appendix C, Provides guidance on the usage and behavior of the PIV Card for authentication purposes.
+ Appendix D, Glossary of Terms and Acronyms, describes the vocabulary and textual representations used in the document.
+ Appendix E, References, lists the specifications and standards referred to in this document.

1.6 Migration Strategy

This document provides two card specifications: 1) Transitional Card Specification as described in Section 2; 2) FIPS 201 PIV–II Card Specification as described in the remainder of this document. Section 2 is a PIV profile derived from the Government Smart Card Interoperability Specification, Version 2.1(NISTIR 6887). This PIV profile is optional, and is presented as one possible path that agencies with substantial quantities of existing GSC-IS v2.1 [10] based smart card deployments may choose to follow during the transition to PIV–II card deployment. All agencies must ultimately comply with the Sections 6 and 7 in accordance with the schedule provided by the Office of Management and Budget(OMB). Full PIV–II deployment is therefore the endpoint of each agency's transition plan.

Agencies may elect to implement the transitional specification as part of a migration path to the mandatory PIV–II specification. This approach may be favorable to agencies who have already deployed substantial quantities of GSC-IS v2.1 based smart cards. Agencies that have not deployed such systems may elect to move directly to deployment of PIV–II systems as they do not have a requirement for backward compatibility with an installed base of GSC-IS v2.1 legacy systems.

The optional migration path offered from Section 2 to the PIV–II specifications is based on continuity of the PIV data model across the two. However, agencies should clearly understand that the card and client interfaces in these two specifications differ. Specific considerations in this migration are highlighted below:

+ Section 2 presents a subset of the dual GSC-IS v2.1 card edge interfaces. PIV–II presents a unified card edge interface that is technology independent and compliant with existing international standards.
+ Card management is provided in PIV–II. A unified and interoperable solution between issuing domains is not achieved when implementing transitional card specification since card management is not addressed in the Section 2.

1.7 PIV Data Model

The PIV data model for SP 800-73 is constructed according to GSC-IS v2.1 specifications. Table 1 – SP 800-73 Data Model Containers defines a high level view of the data model. Each container is labeled as Mandatory or Optional. Mandatory data elements are common and required in both transitional and PIV-II specifications for PIV compliance.

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 0x00000xxxx	ContainerID	Access Rule	Contact / Contactless	M/O
Card Capability Container	0xDB00	Read always	Contact	Mandatory
Card Holder Unique Identifier	0x3000	Read always	Contact & Contactless	Mandatory
X.509 Certificate for PIV Authentication	0x0101	pkiCompute PIN	Contact	Mandatory
Card Holder Fingerprint I	0x6010	PIN	Contact	Mandatory
Card Holder Fingerprint II	0x6011	PIN	Contact	Mandatory
Printed Information	0x3001	PIN	Contact	Optional
Card Holder Facial Image	0x6030	PIN	Contact	Optional
X.509 Certificate for Digital Signature	0x0100	pkiCompute PIN Always	Contact	Optional
X.509 Certificate for Key Management	0x0102	pkiCompute PIN	Contact	Optional
X.509 Certificate for Card Authentication	0x0500	Asymmetric - pkiCompute Always Symmetric - see CCC or CHUID	Contact & Contactless	Optional
Security Object	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 v2.1 specification. It supports minimum capabilities for lookup on data model and application information.

The data model shall be identified by data model number “0x05”. 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 X.509 Certificate for PIV Authentication

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

1.8.3 Card Holder Unique Identifier

The CHUID buffer is defined in accordance with the [7].

The FASC-N is defined to concatenate the “System Code || Credential Number” enabling a credential number space of 9,999,999,999 credentials.

The Global Unique Identification Number (GUID) option is now specified as mandatory and defined as an issuer assigned IPv6 address. If the issuer does not assign this value, it shall be coded as all zeros. The GUID enables migration away from the FASC-N into a robust numbering scheme for all issued credentials.

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 PIV Card Authentication Key.

The Expiration Date is mapped to the RFU tag 0x35, keeping that within the existing scope of the PACS Implementation Guidance, PACS v2.2 [7] specification.

The CHUID is signed in accordance with FIPS 201. The Authentication Key Map is not signed as it can be modified by local PACS systems.

The CHUID is specified as common between the contact and contactless chips. For dual chip implementations, the CHUID is copied in its entirety between the two chips.

1.8.4 Fingerprints

The fingerprint buffers specify the primary and secondary fingerprints in accordance with the FIPS 201. The 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 the Technical Report "PKI for Machine Readable Travel Documents offering ICC read-only access" in accord with ICAO 9303 for MRTD. Tag "0xBA" is used to map the ContainerIDs in the PIV data model to the 16 Data Groups specified in the MRTD. This enables the security object to be fully compliant for future activities with identity documents.

The digital signature key used by the issuer shall be the same key as used to sign the CHUID. The signature field of the CHUID contains the issuer certificate, enabling trust chain validation.

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

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 Card Holder Facial Image

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 X.509 Certificate for Digital Signature

This key and certificate supports Digital Signature for non-repudiation. The PKI compute 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 X.509 Certificate for Key Management

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

1.9.5 X.509 Certificate for Card Authentication

This key and certificate supports PIV Card Authentication for device to device authentication purposes. It does not require cardholder consent to use the key. The access rule for PKI compute functions is "Always". The symmetric key option specified in FIPS 201 is provided by the CHUID in accordance with PACS v2.2.

2. Transitional Card Interface

2.1 PIV Application Programming Interface

2.1.1 Basic Services Interface

This chapter defines the BSI services provided to a PIV application. The following specification is compliant with NISTIR 6887 unless otherwise specified.

The functions listed hereafter are a subset of NISTIR 6887 and required by the PIV application use cases defined in this document:

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

If a PIV client application relies on a middleware, it must be developed according to SP 800-73.

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:

+ Reference to the supported card edge specifications

+ PIV Data Model OID
+ List of mandatory Application with their AID
+ Reference to the mandatory subset of the PIV data model for each application (Object Ids and Tags)
+ 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 (VM or 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	Card Edge Versions Supported (contact)	Card Edge Versions Supported (contactless)
0x00	FULL PIV on VM Card	PIV Data Model OID	ISO 7816-4 aligned GSCIS2.1 VM (see section 9)	GSCIS2.1 Appendix G
0x80	FULL PIV on FS Card	PIV Data Model OID	(see section 9)	GSCIS2.1 Appendix G

2.2.1 PIV Objects Naming Structure

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

Each card application 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 APDU commands available on that object.
- GSC-IS Object ID: Identifies the Container or Object to Select.
- Application ID or PIX.

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

At the BSI level, the objects are referenced by 7 bytes GSCIS Object AID (Issuer RID || GSCIS Object ID).

At the door reader or other PIV applications. The objects are referenced by 2 bytes GSCIS Object IDs.

2.2.2 Mapping mechanisms

The CCC CardApplicationURL is used to lookup the GSCIS Object ID, and constructs the corresponding Application AID 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 v2.1-based systems.

The PIV Application also requires a contactless interface. The contactless command interface is compliant with NISTIR 6887 Appendix G. In both cases, the contactless interface relies on the data model Object IDs and Tags defined in this 800-73 specification that supersedes Appendix G (The CHUID ObjectID/EF is 0x3000). 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 for Contact Interface

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 File System 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 3 and 4.

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

Table 3 – VM Card Commands

Type	Name	Interface
Commands for common interface	SELECT APPLET/ SELECT OBJECT	Contact Only
	GET RESPONSE	Contact Only
Card Platform Commands for Common Interface	READ BUFFER	Contact Only
Commands for Authentication	VERIFY	Contact Only
	PRIVATE SIGN / DECRYPT	Contact Only

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	Interface
Commands for common interface	SELECT	Contact/Contactless
	GET RESPONSE	Contact Only
Card Platform Commands for Common Interface	READ BINARY	Contact/Contactless
Commands for Authentication	VERIFY	Contact Only
	MANAGE SECURITY ENVIRONMENT	Contact Only
	PERFORM SECURITY OPERATION	Contact Only

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 to 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 applet instance (and implicitly the default object).
- 02h for selecting a Card Object.

Data field sent in the command message

In the case of instance selection, the data field contains the instance AID. In the case of Object selection, the data field contains the Card Object ID (OID).

Response Message

Data field returned in the response message

For selecting a card object, the response message is null but 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

This APDU is used to read smart card results available from the completion of the previously executed APDU. GET RESPONSE is usually used to read extended results. This command is applicable to the T=0 transmission protocol.

Command Message

CLA	0x00
INS	0xC0
P1	0x00
P2	0x00
L_c	Empty
Data Field	Empty
L_e	Number of bytes to read in response

Response Message

Data Field returned in the Response Message

If the APDU result indicates success, Le number of bytes will be available to read from the smart card.

Processing State returned in the Response Message

SW1	SW2	Meaning
61	XX	Normal processing, XX still available to read with subsequent Get Response
62	81	Part of returned data may be corrupted
67	00	Wrong length (incorrect L_e field)
6A	86	Incorrect parameters P1-P2
6A	88	Referenced data not found
6C	XX	Wrong length (wrong L_e field; XX indicates the exact length)
90	00	Correct execution

2.3.3.1.3 READ BUFFER APDU

This command allows reading all or part of a buffer.

Command Message

CLA	0x80
INS	0x52
P1	Reference Control Parameter P1
P2	Reference Control Parameter P2
Lc	0x01 + 0x01 = 0x02
Data Field	Buffer type (1 byte value) followed by the data length to read (1 byte value)
Le	Empty

Reference control parameter P1/P2

The reference control parameters P1 and P2 are used to store the offset from which data are to be read (P1 → MSB, P2 → LSB). For example, an offset of 102 (66h) will be encoded using: P1=00h & P2=66h.

Data field sent in the command message

The data field shall be used to indicate which buffer is to be read.

The possible values are:

0x01: T-buffer
0x02: V-buffer

Response Message

Data field returned in the response message

The data field in the response message corresponds to the data read from the smart card, according to the P1, P2 parameters (offset indicating from where to read data) or empty if GET RESPONSE command is required to receive data read from the smart card.

Processing state returned in the response message

SW1	SW2	Meaning
67	00	Invalid command data length
6A	86	Wrong P1/P2 (Try to update data out of the buffer)
6A	88	No corresponding buffer (invalid Buffer Type)
61	XX	for “normal processing, XX bytes of data is read and available for a subsequent Get Response”

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	Key Reference Identifier
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

The parameter P2 indicates the key version. A key version set to 0x00 corresponds to the default key.

Interoperability Note: Effective PIN length and Pin Padding (Addition to NISTIR 6887)

The effective PIN Length (ie. length of data field of VERIFY PIN) must be 8. In all PIV Cards, the PIN is always right padded with 0xFF to the effective pin length of 8 bytes.

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 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.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Meaning
0	X	X	X	X	X	X	X	No more block to follow
1	X	X	X	X	X	X	X	More blocks to follow

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, Section 5.1.1

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

2.4 General Status Conditions

See NISTIR 6887 for General Status Conditions.

3. Concepts and Constructs

Special Publication 800-73 defines two interfaces to an integrated circuit card that contains the Personal Identity Verification card application: a high-level PIV client-application programming interface (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 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 v2.1[10] and in Section 2 described above.

This unification is accomplished by adopting the object-oriented model of computation of the GSC-ISv2.1 virtual machine card edge and realizing its technical details using the data structures and operations found in the international integrated circuit card standards defining the GSC-ISv2.1 file system card edge [1]. This brings the PIV Card application into conformance with those standards with minimal impact on existing GSC-ISv2.1 deployments.

As a result of this unification, the behavior of the PIV card application and the client-applications accessing it is independent of the integrated circuit card platform on which the PIV card application is installed.

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
+ OIDs of the NIST personal authentication arc

+ 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* [2].

A data object whose data content is coded as a BER-TLV data structure [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 integrated circuit card. 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 *application identifier* (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 Proprietary Identifier eXtension (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 *application identifier* (AID) of the *Personal Identity Verification card application* (PIV card application) shall be

'A0 00 00 xx xx 00 00 10 00 01 00'

The AID of the PIV card application consists of the NIST RID ('A0 00 00 xx xx') 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 Applications for Interoperable Use

An integrated circuit card containing the PIV card application may contain other card applications.

An additional card application may be designed and managed in such a way that it is available for use by others. Such a card application is called a *card application for interoperable use*.

An additional card application may be designed and managed in such a way that it is not available for general use. Such a card application is called an *organization-specific card application*. Organization-specific card applications are not intended for global use and the specification for their use may not be available.

3.5 Security Architecture

The security architecture of an integrated circuit card 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 integrated circuit card 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, facial image, and voice recognition. Demonstration of knowledge of the PIN is the authentication protocol for the first security status indicator. Comparison of the facial image with the face of 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 facial image) OR (voice recognition)).

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 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 personal identification number (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 padding bytes shall to be appended to the actual PIN. 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'

Provision of a biometric measurement is another means by which an individual can authenticate themselves to the PIV card application. The format of biometric measurements used by the PIV card application is given in [5].

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 an application identifier and there shall always be a currently selected application.	PIV Platform
Application security status	Yes	Contains security status indicators local to a particular card application.	PIV Card Application

4. PIV Data Objects for Interoperable Use

A PIV card application shall contain six mandatory data objects and give optional data objects 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
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 PIV Card Authentication

Table 6 lists the ASN.1 object identifiers and GSCIS object identifiers 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 the PIV Card Application Capabilities Description, the NIST RID ('A0 00 00 xx xx') shall be used and the card application type shall be set to '00'.

Table 7 lists the access control rules of the eleven PIV card application data objects for interoperable use. See Table 13 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	GSC-IS Object Identifier	M/O
Card Capability Container	2.16.840.1.101.3.7.1.219.0	'DB00'	M
Card Holder Unique Identifier	2.16.840.1.101.3.7.2.48.0	'3000'	M
X.509 Certificate for PIV Authentication	2.16.840.1.101.3.7.2.1.1	'0101'	M
Card Holder Fingerprint I	2.16.840.1.101.3.7.2.96.16	'6010'	M
Card Holder Fingerprint II	2.16.840.1.101.3.7.2.96.17	'6011'	M
Printed Information	2.16.840.1.101.3.7.2.48.1	'3001'	O
Card Holder Facial Image	2.16.840.1.101.3.7.2.96.48	'6030'	O
X.509 Certificate for Digital Signature	2.16.840.1.101.3.7.2.1.0	'0100'	O
X.509 Certificate for Key Management	2.16.840.1.101.3.7.2.1.2	'0102'	O
X.509 Certificate for PIV Card Authentication	2.16.840.1.101.3.7.2.5.0	'0500'	O
Security Object	2.16.840.1.101.3.7.2.144.0	'9000'	M

Table 7 — Access Control Rules of the PIV Data Objects

Data Object for Interoperable Use	CONTACT INTERFACE		CONTACTLESS INTERFACE
	READ	UPDATE	READ
Card Capability Container	Always	PIV Card Application Administrator	Never
Card Holder Unique Identifier	Always	PIV Card Application Administrator	Always
X.509 Certificate for PIV Authentication	Card Holder Global PIN or Card Holder PIV Card Application PIN	PIV Card Application Administrator	Never
Card Holder Fingerprint I and II	Card Holder Global PIN or Card holder PIV Card Application PIN	PIV Card Application Administrator	Never
Printed Information	Card Holder PIV Card Application PIN	PIV Card Application Administrator	Never
Card Holder Facial Image	Card Holder Global PIN or Card Holder PIV Card Application PIN	Card Holder PIV Card Application PIN and PIV Card Application Administrator	Never
X.509 Certificate for Digital Signature	Card Holder PIV Card Application PIN for every use	PIV Card Application Administrator	Never
X.509 Certificate for Key Management	Card Holder PIV Card Application PIN	PIV Card Application Administrator	Never
X.509 Certificate for Card Authentication	Card Holder Global PIN or Card Holder PIV Card Application PIN	PIV Card Application Administrator	Never
Security Object	Always	PIV Card Application Administrator	Never

Updates of any of the eleven data objects for interoperable use may never be made on the contactless interface.

FIPS 201 requires that the PIV card application exhibit different behavior on the contact and contactless interfaces. 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.

Detailed descriptions of these data objects are provided in the SP 800-73 Data Model document.

5. 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 8 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 8 — Cryptographic Algorithm Identifiers

Algorithm Identifier	Algorithm – Mode	Reference Data Length	Message Padding	M/O
'00'	2 Key Triple DES – ECB	16 bytes	NIST SP 800-67	M
'01'	2 Key Triple DES – CBC	16 bytes	NIST SP 800-67	M
'02'	3 Key Triple DES – ECB	24 bytes	NIST SP 800-67	O
'03'	3 Key Triple DES – CBC	24 bytes	NIST SP 800-67	O
'04'	RFU			
'05'	RFU			
'06'	RSA	1024 bits	PKCS #1	M
'07'	RSA	2048 bits	PKCS #1	O
'08'	AES-128 – ECB	24 bytes	FIPS PUB 197	O
'09'	AES-128 – CBC	24 bytes	FIPS PUB 197	O
'0A'	AES-192 – ECB	36 bytes	FIPS PUB 197	O
'0B'	AES-192 – CBC	36 bytes	FIPS PUB 197	O
'0C'	AES-256 – ECB	48 bytes	FIPS PUB 197	O
'0D'	AES-256 – CBC	48 bytes	FIPS PUB 197	O
'0E'	ECC – F2m	163 bits	FIPS PUB 186-2	O
'0F'	ECC – F2m	233 bits	FIPS PUB 186-2	O
'10'	ECC – F2m	283 bits	FIPS PUB 186-2	O
'11'	ECC – Fp-192	192 bits	FIPS PUB 186-2	O
'12'	ECC – Fp-224	224 bits	FIPS PUB 186-2	O
'13'	ECC – Fp-256	256 bits	FIPS PUB 186-2	O

The message padding referred to as PKCS #1 message padding shall be the RSASSA-PSS signature scheme as defined in PKCS #1, Version 2.1.

5.2 Application Property Template

Upon selection, the PIV card application shall return the application property template described in Table 9.

Table 9 — 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 7.
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 10 — Data Objects in a Coexistent Tag Allocation Authority Template (Tag '78')

Description	Tag	M/O	Comment
Application identifier	‘4F’	M	The PIV card application identifier; viz. 'A0 00 00 xx xx 00 00 10 00 01 00 00 00 00 00 00’

5.3 Authenticator

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

Table 11 — Data Objects in an Authenticator Template (Tag '67')

Description	Tag	M/O	Comment
Access mode	‘80’	M	Indicates GET (‘01’) or PUT (‘02’)
Reference data	‘81’	M	E.g. the PIN value or challenge response
Key reference	‘83’	M	See Table 13.

5.4 Connection Description

The connection description BER-TLV used on the PIV client-application programming interface shall have the structure described in Table 12.

Table 12 — Data Objects in a Connection Description Template (Tag '7F81')

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 81 0C 82 04 41 63 6D 65 91 04 81 06 0D 17' describes a connection to the "Acme" card reader at Internet address 129.6.13.23. As another example, '7F 81 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 below, 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 81 04 81 00 90 00' would request 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 13 defines the key reference values that shall be used on the PIV interfaces. All other PIV card application key reference values are reserved for future use.

Table 13 — PIV Card Application Authentication Algorithms and Key References

Algorithm Identifier	Key Reference Value	Key Reference Name	Authenticatable Entity	Security Status	Retry Reset Value
N/A	'00'	Card Holder Global PIN	Card Holder	Global	Platform Specific
N/A	'80'	Card Holder PIV Card Application PIN	Card Holder	Application	'03'
'06'	'9A'	PIV Card Application Validation Key	PIV Card Application Provider	Application	N/A
'00'	'9B'	PIV Card Application Administration Key	PIV Card Application Administrator	Application	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 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 14. The description of individual client-application programming interface entry points or card commands provide additional information for interpreting the status words they return.

Table 14 — Status Words

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available
'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
'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 object identifier (OID) from the NIST personal verification arc and a two-byte object identifier that is backward compatible with GSC-ISv2.1 [10]. These object identifier assignments are given in Table 2 above.

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 two-byte object identifier. For example, the CHUID is identified on the card command interface to the PIV card application by the two-byte identifier '3000'.

6. PIV Client-Application Programming Interface

Table 15 lists the entry points on the PIV client-application programming interface.

Table 15 — 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	**pivSign**
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 integrated circuit card.

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 integrated circuit card. If FALSE and the connection is established then the calling client-application has exclusive access to the integrated circuit card.

connectionDescription A connection description data object (tag '7F 81'). See Table 12.

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 integrated circuit card 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 integrated circuit card 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_CARD_APPLICATION_NO_LONGER_AVAILABLE
PIV_INVALID_CARD_HANDLE
```

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 6.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_APPLICATION_NOT_FOUND
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

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,
   OUT sequence of byte applicationProperties
);
```

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.
applicationProperties	Properties of the card application to which the client-application has been connected.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_APPLICATION_NOT_FOUND
PIV_AUTHENTICATOR_MALFORMED
PIV_AUTHENTICATION_FAILURE
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

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_APPLICATION_NO_LONGER_AVAILABLE
```

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 undefined.

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_NO_CURRENT_CONTEXT
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

6.3 Entry Points for Cryptographic Operations

6.3.1 pivSign

Purpose: Create a digital signature.

Prototype:

```
status_word pivSign(
   IN handle               cardHandle,
   IN byte                 algorithmIdentifier,
   IN byte                 keyReference,
   IN sequence of byte     bytesToBeSigned,
   OUT sequence of byte    digitalSignature
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
algorithmIdentifier	Identifier of the cryptographic algorithm to be used to create the digital signature. See Table 5.
keyReference	Identifier of key reference to be used to create the digital signature. See Table 10.
bytesToBeSigned	Sequence of bytes, for example a hash, for which a digital signature is to be created.
digitalSignature	The digital signature.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_KEY_REFERENCE
PIV_REFERENCE_DATA_NOT_FOUND
PIV_BYTES_TO_BE_SIGNED_MALFORMED
PIV_INSUFFICIENT_CARD_RESOURCE
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

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_OFFSET_BEYOND_END_OF_DATA_CONTENT
PIV_INSUFFICIENT_CARD_RESOURCE
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

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 integer             keyReference,
   IN integer             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 21.

publicKey	BER-TLV data objects defining the public key of the generated key pair. See Table 22.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_KEY_REFERENCE
PIV_GENERATED_KEY_PAIR_INCOMPATIBLE_WITH_EXISTING_KEY_PAIR
PIV_UNSUPPORTED_CRYPTOGRAPHIC_MECHANISM
PIV_INSUFFICIENT_CARD_RESOURCE
PIV_CARD_APPLICATION_NO_LONGER_AVAILABLE
```

7. PIV Card Application Card Command Interface

The Table 16 lists the card commands surfaced by the PIV card application at the card edge of the integrated circuit card containing it. All PIV card application card commands shall be supported by a PIV card application.

Table 16 — PIV Card Application Card Commands

Type	Name	Contact Interface	Contactless Interface	Security Condition for Use
PIV Card Application Card Commands for Data Access	**SELECT APPLICATION**	Yes	Yes	Always
	GET DATA	Yes	Yes	Data Dependent. See Table 3.
PIV Card Application Card Commands for Authentication	**VERIFY PIN**	Yes	No	Always
	CHANGE REFERENCE DATA	Yes	No	Card Holder PIV Card Application PIN
	RESET RETRY COUNTER	Yes	No	PIV Card Application Administrator and Card Holder Biometric
	GENERAL AUTHENTICATE	Yes	See Note	PIV Card Application Administrator
PIV Card Application Card Commands for Credential Initialization and Administration	**PUT DATA**	Yes	No	Data Dependent. See Table 3.
	GENERATE ASYMMETRIC KEY PAIR	Yes	No	PIV Card Application Administrator

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 16.

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

7.1 PIV Card Application Card Commands for Data Access

7.1.1 SELECT APPLICATION Card Command

The SELECT APPLICATION card command sets the currently selected application.

If the currently selected application when the SELECT APPLICATION command is given is not the application whose AID is in the data field of the SELECT APPLICATION then the currently selected application shall be deselected and all application security status indicators shall be set to FALSE.

If the currently selected application when the SELECT APPLICATION command is given is the application whose AID is in the data field of the SELECT APPLICATION the setting of all security status indicators shall be unchanged.

If there are one or more card applications the leading bytes of whose application identifiers match the provided application identifier, then the most recent card application of these shall be selected and the complete application identifier of the application returned in the response. As the PIX of the AID of the PIV card application contains an encoding of the version of the PIV card application this enables a client-application to use a truncated PIV card application AID in the SELECT APPLICATION command and determine the version of the PIV card application if any on the integrated circuit card from the returned application property template.

There shall be at most one PIV card application on any integrated circuit card.

Command Syntax

CLA	'00'
INS	'A4'
P1	'04'
P2	'00'
L_c	Length of application identifier
Data Field	Application identifier (AID)
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	'06'
Data Field	See Table 17.
L_e	Number of data content bytes to be retrieved.

Table 17 — Data Objects in the Data Field of the GET DATA Card Command

Name	Tag	M/O	Comment
Tag list	'5C'	M	Tag or object identifier of the data object to be retrieved. For object identifiers, see Column 3 of Table 2.

Response Syntax

Data Field	BER-TLV with the tag '53' or '73' containing in its value field the requested data content. The tag is '73' if the data object is a BER-TLV data object. Otherwise, the tag is '53'.
SW1-SW2	Status word

SW1	SW2	Meaning
'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 PIN Card Command

The VERIFY PIN 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.

Command Syntax

CLA	'00'
INS	'20'
P1	'00'
P2	Key reference. See Table 13.
L_c	'08'
Data Field	Authentication data; i.e. PIN
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'00'	Verification failed
'63'	'CX'	Verification failed, X indicates the number of further allowed retries
'69'	'83'	Authentication method blocked
'6A'	'88'	Reference data 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.

Command Syntax

CLA	'00'
INS	'24'
P1	'00'
P2	Key reference. See Table 13.
L_c	'10'
Data Field	Verification data followed without delimitation by new reference data
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'00'	Verification failed
'69'	'83'	Authentication method blocked
'6A'	'88'	Reference data not found
'90'	'00'	Successful execution

7.2.3 RESET RETRY COUNTER Card Command

The RESET RETRY COUNTER card command initiates the comparison of verification data with reset reference data and if this comparison is successful resets the reference data retry counter to its initial value and replaces the reference data with new reference data.

Command Syntax

CLA	'00'
INS	'2C'
P1	'00'
P2	Key reference. See Table 13.
L_c	'10'
Data Field	Verification data followed without delimitation by new reference data
L_e	Empty

Response Syntax

SW1	SW2	Meaning
'63'	'00'	Verification failed
'63'	'CX'	Verification failed, X indicates the number of further allowed retries
'69'	'83'	Authentication method blocked
'6A'	'88'	Reference data not found
'90'	'00'	Successful execution

7.2.4 GENERAL AUTHENTICATE Card Command

The GENERAL AUTHENTICATE card command performs an authentication protocol using the data provided in the data field of the command and returns the result of the authentication protocol 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.

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'
INS	'87' or '97'
P1	Algorithm reference
P2	Key reference
L_c	Length of data field
Data Field	See Table 18.
L_e	Absent or length of expected response

Table 18 —Data Field of the GENERAL AUTHENTICATE Card Command

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
'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 replaces the data content of a single data object in the PIV card application.

Command Syntax

CLA	'00'
INS	'DB'
P1	'3F'
P2	'FF'
L_c	Length of data field
Data Field	See Table 19.
L_e	Empty

Table 19 — 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 2.
Data	'53' or' 73'	M	Data with tag '53' is an unstructured byte sequence. Data with tag '73' is a BER-TLV.

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.

Command Syntax

CLA	'00'
INS	'47'
P1	'00'
P2	Key reference to be assigned to the generated asymmetric key pair.
L_c	Length of data field
Data Field	Control reference template. See Table 20.
L_e	Length of public key of data object template

Table 20 —Data Field of the GENERATE ASYMMETRIC KEY PAIR Command

Name	Tag	M/O	Description
Cryptographic mechanism identifier	'80'	M	See Table 21.
Parameters	'81'	C	Specific to the cryptographic mechanism

Table 21 — Cryptographic Mechanism Identifiers

Cryptographic Mechanism Identifier	Description	M/O	Parameters
'00'	RFU		
'01'	RSA 1024	M	None
'02'	RSA 2048	O	None
'03'	RSA 3072	O	None
'04'-'10'	RFU		
'11'	ECC – F2m-163	O	None
'12'	ECC – F2m-233	O	None
'13'	ECC – F2m-283	O	None
'14'	ECC – Fp-192	O	None
'15'	ECC – Fp-224	O	None
'16'	ECC – Fp-256	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 22.
SW1-SW2	Status word

Table 22 —Data Field of the GENERATE ASYMMETRIC KEY PAIR Response

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
'69'	'82'	Security status not satisfied
'6A'	'80'	Incorrect parameter in command data field; e.g. unrecognized cryptographic mechanism
'6A'	'86'	Incorrect parameter P1; 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 RID for the PIV Data Model can be 0xA0 00 00 xx xx to allow for agency specific number as described in Section 1. The RID for the Card Capability Container must be 0xA0 00 00 01 16. The registered PIV data model number is 0x05, and the data model version number is 0x01.

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	2609	Always Read	Contact and Contactless	M
X.509 Certificate for PIV Authentication	0x0101	1139	pkiCompute - PIN	Contact	M
Card Holder Fingerprint I	0x6010	7000	PIN	Contact	M
Card Holder Fingerprint II	0x6011	7000	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	1139	pkiCompute -PIN Always	Contact	O
X.509 Certificate for Key Management	0x0102	1139	pkiCompute - PIN	Contact	O
X.509 Certificate for Card Authentication	0x0500	1139	Asymmetric – pkiCompute – PIN Symmetric – See CCC or CHUID	Contact and Contactless	O
Security Object	0x9000	600	Always Read	Contact	M

Card Capabilities Container	0xDB00	Always Read	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Card Identifier	0xF0	Fixed	21
Capability Container version number	0xF1	Fixed	1
Capability Grammar version number	0xF2	Fixed	1
Applications CardURL	0xF3	Variable	128
PKCS#15	0xF4	Fixed	1
Registered Data Model number	0xF5	Fixed	1
Access Control Rule Table	0xF6	Fixed	17
CARD APDUs	0xF7	Fixed	0
Redirection Tag	0xFA	Fixed	0
Capability Tuples (CTs)	0xFB	Fixed	0
Status Tuples (STs)	0xFC	Fixed	0
Next CCC	0xFD	Fixed	0
Extended Application CardURL	0xE3	Fixed	48

Card Capabilities Container		0xDB00	Always Read
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Security Object Buffer	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	0x3D	Variable	512
Issuer Asymmetric Signature	0x3E	Variable	2048
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	1100
CertInfo	0x71	Fixed	1
MSCUID	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Card Holder Fingerprint		0x6010	PIN
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Fingerprint	0xBC	Variable	7000
Error Detection Code	0xFE	LRC	0

Card Holder Fingerprint		0x6011	PIN
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Fingerprint	0xBC	Variable	7000
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	1100
CertiInfo	0x71	Fixed	1
MSCUID	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	1100
CertiInfo	0x71	Fixed	1
MSCUID	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

X.509 Certificate for Card Authentication	0x0500	Asymmetric – pkiCompute -PIN Symmetric – See CCC / CHUID	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes**
Certificate	0x70	Variable	1100
CertiInfo	0x71	Fixed	1
MSCUID	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	500
Error Detection Code	0xFE	LRC	0

Appendix B—Use of the GENERAL AUTHENTICATE Card Command

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; viz. 2 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 23 shows the GENERAL AUTHENTICATE card commands sent to the PIV card application to realize this particular challenge/response protocol.

Table 23 — 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 12 7C 10 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 6 and 10.
	'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 Card Application Validation Key (OID 2.16.840.1.101.3.7.1.1.2.2.1) 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 24 shows the GENERAL AUTHENITCATE card commands sent to the PIV card application to realize the validation of the PIV card application.

Table 24— Validation of the PIV Card Application Using GENERA 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 6 and 10.
	‘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.

B.3 Contactless Mutual Authentication

Contactless card style mutual authentication is initiated by the terminal and consists of retrieving a short (e.g. 5-byte) encrypted nonce (random byte sequence) from the card indicating in the P1 and P2 fields the cryptographic algorithm and key reference respectively to be used in the authentication protocol. The response from the card is the encryption of a first nonce according to P1 and P2.

The terminal then sends a decryption of the first encrypted nonce together with an unencrypted second nonce to the card. The response from the card is the encryption of the second nonce according to the previously indicated cryptographic algorithm and key reference.

This procedure is summarized in the Table 25. If N represents the number of bytes in an encrypted or unencrypted nonce then the total number of bytes transmitted in the contactless authentication protocol is 4N+26.

Table 25 — Mutual Authentication Using GENERAL AUTHENTICATE

Command	Response	Bytes	Comment
'00 87 06 9A 04 7C 02 80 00'		9	Terminal requests a witness from the PIV card application using a the reference data associated with a specified key reference and a specified algorithm. See Tables 6 and 10.
	'7C' 'N+2' '80' 'N' {Encryption of card nonce}	N+4+2	PIV card application responds with the encryption of a nonce using the specified reference data and algorithm.
'00 87 06 9A' '2N+6' '7C' '2N+4' '80' 'N' {Decryption of card nonce} '81' 'N' {Terminal nonce}		11+2N	Terminal returns decryption of the encrypted nonce as a witness along with another nonce.
	'7C' 'N+2' '80' 'N' {Encryption of terminal nonce}	N+4+2	PIV card application verifies the witness and then responds with the encryption of the nonce sent by the terminal using the specified reference data and algorithm.

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)
  - 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 thee PIV Card, and is illustrated in Figure B-1.

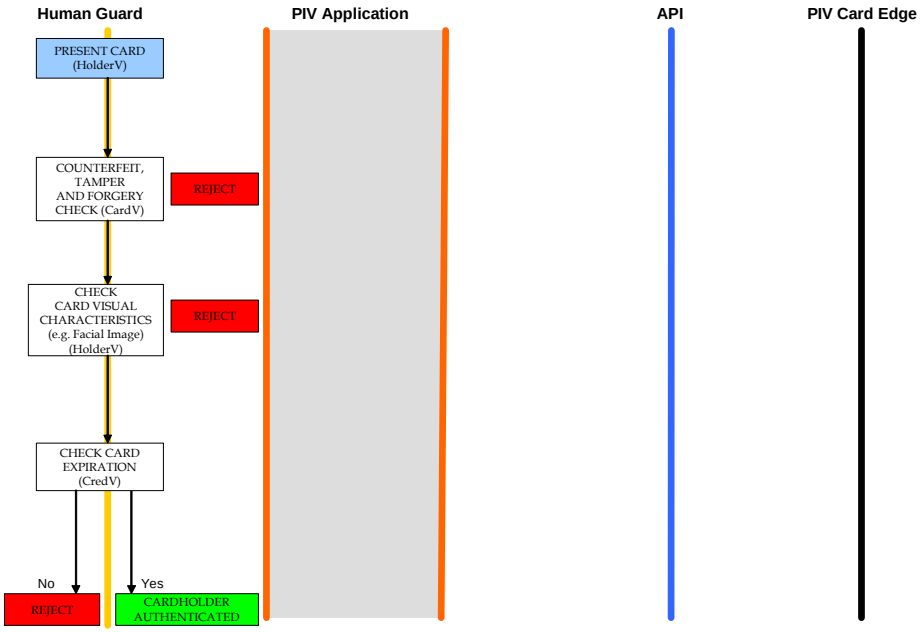

Figure B-1: Authentication using PIV Visual Credentials

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 B-2.

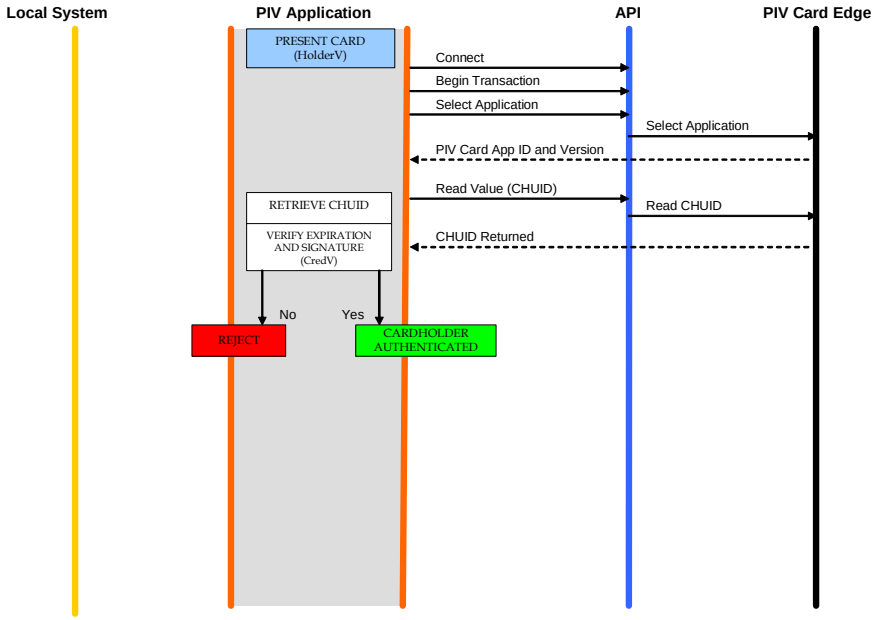

Figure B-2: Authentication using PIV CHUID

C.1.3 Authentication using PIV Biometrics

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

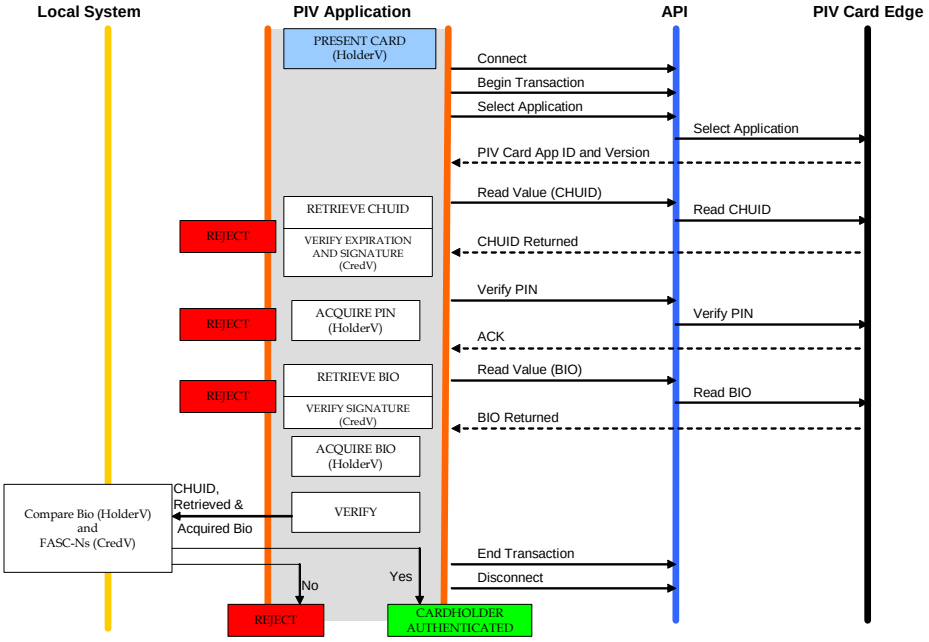

Figure B-3: Authentication using PIV Biometrics

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 B-4.

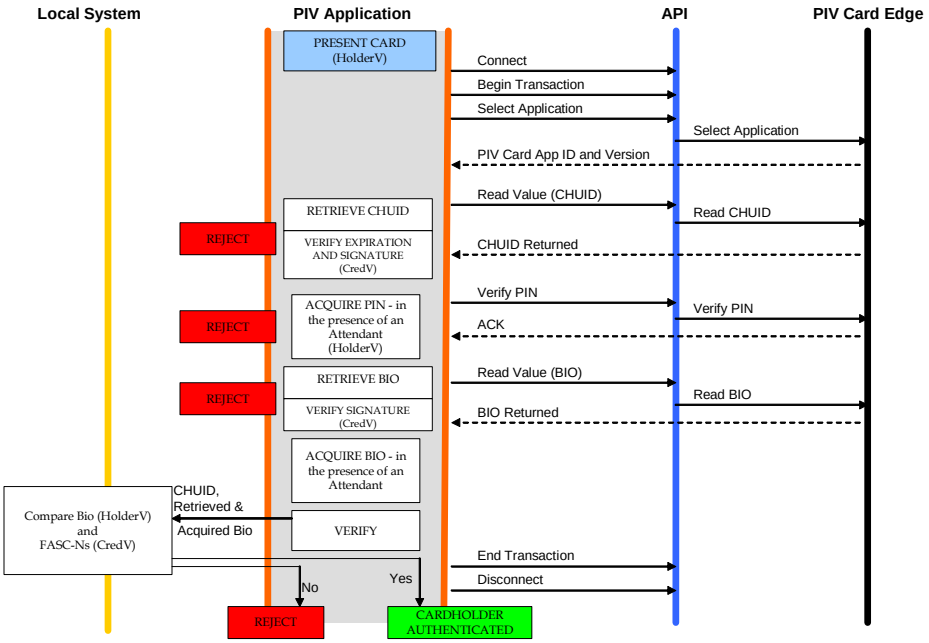

Figure B-4: Authentication using PIV Biometrics (Attended)

C.1.4 Authentication using PIV Authentication Key

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

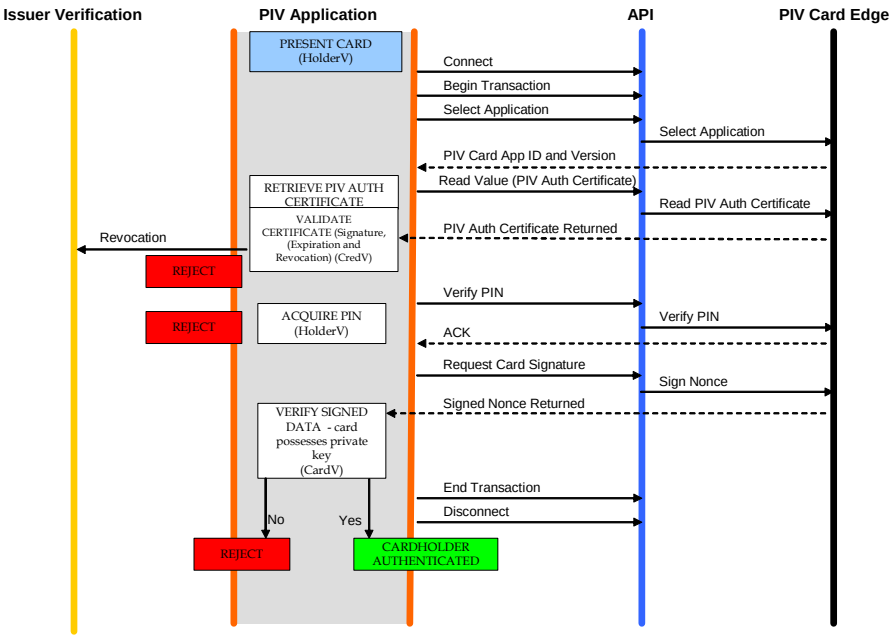

Figure B-5: Authentication using PIV Authentication Key

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 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 a 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.
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
BER	Basic Encoding Rules
CBC	Circular Binary Coding
CLA	Class (first) byte of a card command
CHUID	Card Holder Unique IDentifier
DES	Data Encryption Standard
DO	Data Object
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
FASC-N	Federal Agency Smart Credential Number
GUID	Global Unique Identification Number
ICC	Integrated Circuit Card
IFD	Interface Device
INS	Instruction (second) byte of a card command
LSB	Least Significant Bit
MSB	Most Significant Bit
OID	Object IDentifier
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
RFU	Reserved for Future Use
RID	Registered application provider IDentifier
RSA	Rivest, Shamir, Aldeman
SCP	ETSI Smart Card Project
SEIWG	Security Equipment Integration Working Group
SP	Special Publication
SW1	First byte of a two-byte status word
SW2	Second byte of a two-byte status word
TLV	Tag-Length-Value

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 reserved for future use (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 [9].

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] ISO/IEC 9834-7:1998, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Assignment of international names for use in specific contexts.*

[5] NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification*, expected March, 2005.

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

[7] 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.

[8] NIST Federal Information Processing Standards Publication 186-2, *Digital Signature Standard (DSS),* January, 2000.

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

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