

NIST Special Publication 800-73  
Draft

# Interfaces for Personal Identity Verification

# NIST

**National Institute of  
Standards and Technology**

Technology Administration  
U.S. Department of Commerce

## INFORMATION SECURITY

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

*January 31, 2005*



**U.S. Department of Commerce**  
*Donald L. Evans, Secretary*

**Technology Administration**  
*Phillip J. Bond, Under Secretary for Technology*

**National Institute of Standards and Technology**  
*Hratch G. Semerjian, Acting Director*



**NOTE FOR REVIEWERS**

1. NIST has revised this Special Publication 800-73 (SP 800-73) in response to the comments received on the FIPS 201 public draft. The SP 800-73 provides the specifications for interfacing with the Personal Identity Verification (PIV) Card. Please note that the revised SP 800-73 is an abstraction of and compatible with both file system and virtual machine cards. It provides a streamlined, ISO compliant unified card edge independent of the underlying card platform technology.
2. 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.
3. Comments should be submitted to [DraftFips201@nist.gov](mailto: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.
4. The comment period closes at 5:00 EST (US and Canada) on February 14th, 2005. Comments received after the comment period closes will be handled on as-time-is-available basis.

## **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, 48 pages  
(January 31, 2005)**

## **Acknowledgements**

The authors wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Interagency Advisory Board (IAB) for providing detail 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.

## Executive Summary

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) for 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 data from the PIV Card<sup>1</sup> and is a companion document to FIPS 201.

---

<sup>1</sup> 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).

## Table of Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. USE CASES.....</b>	<b>3</b>
2.1 CARD VALIDATION .....	3
2.2 REGISTRATION .....	3
2.3 CARDHOLDER AUTHENTICATION FOR LOGICAL ACCESS .....	3
2.4 CARDHOLDER AUTHENTICATION FOR PHYSICAL ACCESS.....	3
<b>3. CONCEPTS AND CONSTRUCTS.....</b>	<b>4</b>
3.1 DATA OBJECTS .....	4
3.1.1 <i>Data Object Content</i> .....	4
3.2 CARD APPLICATIONS .....	5
3.2.1 <i>Personal Identity Verification Card Application</i> .....	5
3.2.2 <i>Applications for Interoperable Use</i> .....	5
3.3 SECURITY ARCHITECTURE.....	5
3.3.1 <i>Access Control Rule</i> .....	6
3.3.2 <i>Security Status</i> .....	6
3.3.3 <i>Authentication of an Individual</i> .....	6
3.4 CURRENT STATE OF THE PIV INTEGRATED CIRCUIT CARD.....	6
<b>4. DATA OBJECTS FOR INTEROPERABLE USE .....</b>	<b>8</b>
4.1 X.509 CERTIFICATE OF THE CARD VALIDATION KEY .....	8
4.2 CARDHOLDER UNIQUE IDENTIFIER (CHUID) .....	8
4.3 CARDHOLDER BIOMETRIC .....	9
<b>5. DATA TYPES AND THEIR REPRESENTATIONS.....</b>	<b>10</b>
5.1 ACCESS MODE IDENTIFIER.....	10
5.2 ALGORITHM IDENTIFIER .....	10
5.3 AUTHENTICATOR .....	11
5.4 CARD APPLICATION PROPERTY TEMPLATE.....	11
5.5 CONNECTION DESCRIPTION .....	11
5.6 KEY REFERENCES .....	12
5.7 STATUS WORDS .....	12
5.8 OBJECT IDENTIFIERS .....	13
<b>6. PIV CLIENT-APPLICATION PROGRAMMING INTERFACE.....</b>	<b>14</b>
6.1 ENTRY POINTS FOR COMMUNICATION .....	14
6.1.1 <i>pivConnect</i> .....	14
6.1.2 <i>pivAcquireContext</i> .....	15
6.1.3 <i>pivReleaseContext</i> .....	16
6.1.4 <i>pivDisconnect</i> .....	16
6.2 ENTRY POINTS FOR DATA ACCESS.....	16
6.2.1 <i>pivSelectCardApplication</i> .....	16
6.2.2 <i>pivGetData</i> .....	17
6.3 ENTRY POINTS FOR CRYPTOGRAPHIC OPERATIONS .....	18
6.3.1 <i>pivSign</i> .....	18
6.4 ENTRY POINTS FOR CREDENTIAL INITIALIZATION AND ADMINISTRATION.....	18
6.4.1 <i>pivPutData</i> .....	18
6.4.2 <i>pivGenerateKeyPair</i> .....	19

**7. PIV CARD COMMAND INTERFACE.....21**

7.1 PIV CARD APPLICATION COMMANDS FOR DATA ACCESS .....21

7.1.1 *SELECT APPLICATION Command*.....21

7.1.2 *GET DATA Command* .....22

7.2 PIV CARD APPLICATION COMMANDS FOR AUTHENTICATION .....23

7.2.1 *VERIFY PIN Command* .....23

7.2.2 *CHANGE REFERENCE DATA Command*.....24

7.2.3 *RESET RETRY COUNTER Command*.....25

7.2.4 *GENERAL AUTHENTICATE Command*.....25

7.3 PIV CARD APPLICATION COMMANDS FOR CREDENTIAL INITIALIZATION AND ADMINISTRATION .....26

7.3.1 *PUT DATA Command* .....26

7.3.2 *GENERATE ASYMMETRIC KEY PAIR Command*.....27

**8. USE CASE DATA FLOWS .....30**

8.1 CARD AUTHENTICATION.....30

8.2 REGISTRATION .....31

8.3 CARDHOLDER AUTHENTICATION FOR PHYSICAL ACCESS.....32

8.4 CARDHOLDER AUTHENTICATION FOR LOGICAL ACCESS .....33

**List of Figures**

Figure 1 — PIV Architecture..... 1

Figure 2 — Data Flow for Card Authentication Use Case.....30

Figure 3 — Data Flow for Registration Use Case.....31

Figure 4 — Data Flow for Cardholder Authentication for Physical Access Use Case .....32

Figure 5 — Data Flow for Cardholder Authentication for Logical Access Use Case .....33

**List of Tables**

Table 1 — State of the PIV Integrated Circuit Card ..... 7

Table 2 — Object Identifiers of the PIV Data Objects for Interoperable Use..... 8

Table 3 — Access Control Rules of the PIV Data Objects for Interoperable Use..... 8

Table 4 — CHUID Data Object ..... 9

Table 5 — Access Mode Identifiers.....10

Table 6 — Algorithm Identifiers .....10

Table 7 — Data Objects in an Authenticator Template (Tag '67') .....11

Table 8 — Data Objects in a Card Application Property Template (Tag '61').....11

Table 9 — Data Objects in a Connection Description Template .....11

Table 10 — PIV Key References .....12

Table 11 — Status Words .....12

Table 12 — Entry Points on PIV Client-Application Programming Interface .....14

Table 13 — PIV Card Application Commands.....21

Table 14 — Data Objects in the Data Field of the GET DATA Command.....23

Table 15 — Data Objects in the Template (Tag '7C') .....26

Table 16 — Data Objects in PUT DATA Data Field.....27

Table 17 — Data Objects in the Template (Tag 'AC') .....28

Table 18 — Cryptographic Mechanism .....28

Table 19 — Data Objects in the Template (Tag '7F49').....28

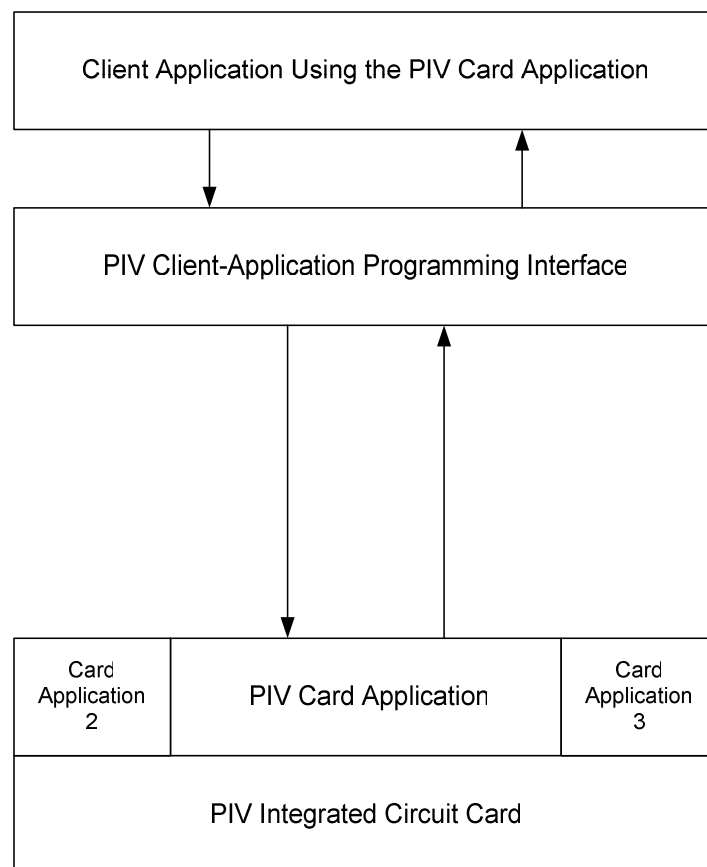
Table 20 — Mutual Authentication Commands .....34



## 1. Introduction

An integrated circuit card is a personal, portable and tamper-resistant data store with cryptographic processing capabilities. The data and the cryptographic processing capabilities are accessed by client applications through interaction with applications on the integrated circuit card.

A client-application program accessing the capabilities of an integrated circuit card sends commands to a card application through the card command interface by means of placing calls on a high-level, task-oriented client-application programming interface. The card commands themselves are executed by the card application inside the integrated circuit card. The results of executing the card commands are returned to client-application program by way of the high-level client-application programming interface.



**Figure 1 — PIV Architecture**

Special Publication 800-73 describes the client-application programming interface and the card command interface for use of the Federal Personal Identity Verification (PIV) integrated circuit card.

Special Publication 800-73 is provided in sufficient technical detail that compliant client-application programs, compliant card applications and compliant integrated circuit cards can be used interchangeably by all information processing systems conforming to this publication.

Special Publication 800-73 is organized as follows:

- + Section 2, Use Cases, describes in general terms the four use cases for which the PIV client-application programming interface and PIV card command interface were required at a minimum to support. The uses of the PIV integrated circuit card and the card applications thereon are not limited to these use cases.
- + Section 3, Concepts and Constructs, describes the model of computation of the PIV card application including the information processing concepts and the data representation constructs.
- + Section 4, Data Objects for Interoperable Use, describes the format and coding of the PIV card for inter-agency and interoperable use as viewed at and accessed on the PIV card edge.
- + Section 5, Data Types and their Representations, provides the details of the data found on the PIV client-application programming interface and PIV 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 Command Interface, describes the card command interface to the PIV card application found on every PIV integrated circuit card.
- + Section 8, Use Case Data Flows, describes the realization of the four use cases of Section 3 in terms of the interfaces described in Section 7 and Section 8.
- + Appendix A, Examples of GENERAL AUTHENTICATE, provides an example of the use of the GENERAL AUTHENTICATE command to do mutual authentication for contactless cards and references a source for other examples.
- + Appendix B, Terms, Acronyms, and Notation, describes the vocabulary and textual representations used in the document.
- + Appendix C, References, lists the specifications and standards referred to in this specification.

## **2. Use Cases**

Four use cases have been identified for the Personal Identification Verification integrated circuit card. These are described in general terms in the following paragraphs. Section 8 below provides the data flows for the realization of these four use cases using the PIV interfaces defined in Sections 6 and 7 below.

### **2.1 Card Validation**

During card validation one verifies that the card being presented is a valid card and that it is in possession of the rightful cardholder.

### **2.2 Registration**

In the case that the card is presented to an entity other than the issuing entity, data on the card may be registered with the entity to which the card has been presented.

### **2.3 Cardholder Authentication for Logical Access**

The individual presenting the card is linked to the card through the use of an authentication protocol supported by the card. An example of such an authentication protocol is the verification by the card of a PIN or password provided by the individual presenting the card.

### **2.4 Cardholder Authentication for Physical Access**

The individual presenting the card is linked to the card through the use of an authentication protocol supported by the card. An example of such an authentication protocol is the verification of a biometric reading acquired from the individual against a biometric template stored on the card.

### 3. Concepts and Constructs

Special Publication 800-73 defines two interfaces to the Personal Identity Verification integrated circuit card: the high-level PIV client-application programming interface (API) and the low-level PIV card command interface (card edge).

The information processing concepts and data constructs on both interfaces are identical and are referred to generically as information processing concepts and data constructs on the *PIV interface* without reference to a particular 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 integrated circuit card. The card command interface is used by software implementing the client-application programming interface.

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 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 where as 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 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*.

##### 3.1.1 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*.

## 3.2 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 6]. 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*.

### 3.2.1 Personal Identity Verification Card Application

The *personal identity verification card application* is a card application that shall be present on every PIV integrated circuit card. The AID of the PIV card application shall be 'A0 xx xx xx xx xx xx xx xx'.

The PIV card application shall be the currently selected application when the card is initially activated or reset and is therefore called the *default application* on the PIV integrated circuit card.

The cardholder unique identity (CHUID) data object is the currently selected application when the PIV card application is selected and it is therefore called the *default data object* in the PIV card application.

The consequence of the previous two paragraphs is that when a PIV integrated circuit card is activated, either on the contact or the contactless interface, the PIV card application shall be the currently selected application and the CHUID is the currently selected data object. Therefore, the CHUID may be immediately read with the first card command sent to the PIV integrated circuit card.

The PIV card application surfaces the commands described in Section 6 on the card command interface.

### 3.2.2 Applications for Interoperable Use

A PIV integrated circuit card may contain card applications in addition to the mandatory PIV card application.

An additional card application may be an *organization-specific card application* and used by only the organization that places the application on the PIV integrated circuit card. Organization-specific card applications are not intended for interoperational use and the specification for their use may not be published.

On the other hand, an additional card application may be designed and managed in such a way that it is available for use by all PIV card programs. Such a card application is called a *card application for interoperable use*.

## 3.3 Security Architecture

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

The operating software in the integrated circuit card applies these security policy representations to all card commands thereby insuring that the card applications implement and enforce the prescribed data security policies.

The following subsections describe the security architecture of the PIV card.

### 3.3.1 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 in Boolean 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.

### 3.3.2 Security Status

Associated with each authenticable entity shall be a Boolean variable called the *security status indicator* of the authenticable entity. The security status indicator of an authenticable entity shall be TRUE if the credentials of the authenticable entity have been authenticated and FALSE otherwise.

All security status indicators shall be set to FALSE when the PIV integrated circuit card is reset.

The successful execution of an authentication protocol by an application shall set the security status indicator of the authenticable entity whose credentials were verified by the protocol to TRUE.

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 indicator status.

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.

### 3.3.3 Authentication of an Individual

Knowledge of a personal identification number (PIN) is a means by which an individual can authenticate themselves to an integrated circuit card. The format of personal identification numbers used on the PIV integrated circuit card is given in [4].

Provision of a biometric reading is another means by which an individual can authenticate themselves to an integrated circuit card. The format of biometric readings used on the PIV integrated circuit card is given in [6].

## 3.4 Current State of the PIV Integrated Circuit Card

The elements of the *state* of a PIV integrated circuit card are listed and described in Table 1.

**Table 1 — State of the PIV Integrated Circuit Card**

<b>State Name</b>	<b>Always Defined</b>	<b>Comment</b>
Currently selected application	Yes	There is always a currently selected application; the default application on the PIV integrated circuit card is the PIV card application
Currently selected data object	No	The currently selected data object can be set by use of the GET DATA card command. A card application can define a default data object that becomes the currently selected data object when the application is selected. The default data object for the PIV card application is the CHUID whose OID is 2.16.840.1.101.3.7.1.1.4.3.1.
Global security status	Yes	Contains security status indicators that span all applications.
Application security status	Yes	Contains security status indicators local to a particular application.

## 4. Data Objects for Interoperable Use

The PIV integrated circuit card shall contain three data objects for interoperable use. These three data objects support the four PIV use cases described in Section 2 above. The three PIV data objects for interoperable use are —

1. X.509 Certificate of the Card Validation Key
2. CHUID
3. Cardholder Biometric

The details of the content, format and encoding of these three data objects is given in Federal Information Processing Standard 201 (FIPS 201). [7]

Table 2 lists the object identifiers of the data objects for interoperable use and Table 3 lists the access control rules of the data objects for interoperable use.

**Table 2 — Object Identifiers of the PIV Data Objects for Interoperable Use**

Data Object for Interoperable Use	OID	BER-TLV Encoding
X.509 Certificate of the Card Validation Key	2.16.840.1.101.3.7.1.1.2.2.1	'06' '0C' '608648016503070101020201'
CHUID	2.16.840.1.101.3.7.1.1.4.3.1	'06' '0C' '608648016503070101040301'
Cardholder Biometric	2.16.840.1.101.3.7.1.1.4.2.1	'06' '0C' '608648016503070101040201'

**Table 3 — Access Control Rules of the PIV Data Objects for Interoperable Use**

Data Object for Interoperable Use	CREATE	READ	UPDATE	DELETE
X.509 Certificate of the Card Validation Key	Card Issuer	Cardholder	Cardholder or Card Issuer	Never
CHUID	Card Issuer	Always	Card Issuer	Never
Cardholder Biometric	Card Issuer	Cardholder	Card Issuer	Never

### 4.1 X.509 Certificate of the Card Validation Key

The technical details of this data object including format, content and encoding are provided in FIPS 201.

### 4.2 Cardholder Unique Identifier (CHUID)

PIV cards shall contain a CHUID as specified in [8], with additions required by FIPS 201. [8] defines the format for the Federal Agency Smart Credential Number (FASC-N), which is the only mandatory element in the CHUID object. FIPS 201 also mandates the asymmetric signature field and defines the format of that field and procedures for calculating signature values. In addition, FIPS 201 mandates an Expiration Date field. The Expiration Date shall be encoded as the American Standard Code for Information

Interchange (ASCII) representation of the year, month and day (commonly referred to as “YYMMDD” format) with no separation characters in a fixed length field of six bytes. For example, an expiration date of February 25, 2005 would be encoded as “050201”. The data format for CHUID is provided in Table 4.

**Table 4 — CHUID Data Object**

<b>Data Element</b>	<b>Tag</b>	<b>Type</b>	<b>Max. Bytes</b>
Buffer Length	EE	Fixed	2
FASC-N (SEIWG-012)	30	Fixed	25
(PACS Reserved Tags)	31-3D		
Asymmetric Signature	3E	Variable	TBD
Expiration Date	40	Fixed	6
Error Detection Code	FE	LRC	1

### 4.3 Cardholder Biometric

The biometric data requirements are provided in FIPS 201 and the technical details of this data object including format, content and encoding are provided in [6].

## 5. Data Types and Their Representations

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

### 5.1 Access Mode Identifier

Table 5 lists the identifiers of the access modes that shall be defined for the data objects on the PIV interface.

**Table 5 — Access Mode Identifiers**

Access Mode	Identifier
Get (Read)	'81'
Put (Write, Update)	'82'
Create	'83'
Delete	'84'

### 5.2 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. For the match algorithm, the reference data length shall be the maximum length of a password or PIN. For the other algorithms, the reference data length shall be the length of a key in bits. Table 6 lists the algorithm identifiers for the cryptographic algorithms that shall be recognized on the PIV interface.

**Table 6 — Algorithm Identifiers**

Algorithm Identifier	Algorithm – Mode	Reference Data Length	Message Padding
'01'	Match	8 bytes	ISO/IEC 9564
'10'	RSA	1024 bits	PKCS #1
'11'	RSA	2048 bits	PKCS #1
'20'	ECC – F2m	163 bits	PKCS #1
'21'	ECC – F2m	233 bits	PKCS #1
'22'	ECC – F2m	283 bits	PKCS #1
'23'	ECC – Fp-192	192 bits	PKCS #1
'24'	ECC – Fp-224	224 bits	PKCS #1
'25'	ECC – Fp-256	256 bits	PKCS #1
'30'	2 key Triple DES – ECB	16 bytes	NIST SP 800-67
'31'	2 key Triple DES – CBC	16 bytes	NIST SP 800-67
'32'	3 key Triple DES - ECB	24 bytes	NIST SP 800-67
'33'	3 key Triple DES - CBC	24 bytes	NIST SP 800-67
'40'	AES-128 - ECB	24 bytes	FIPS PUB 197
'41'	AES-128 - CBC	24 bytes	FIPS PUB 197
'42'	AES-192 – ECB	36 bytes	FIPS PUB 197

Algorithm Identifier	Algorithm – Mode	Reference Data Length	Message Padding
'43'	AES-192 – CBC	36 bytes	FIPS PUB 197
'44'	AES-256 – ECB	48 bytes	FIPS PUB 197
'45'	AES-256 - CBC	48 bytes	FIPS PUB 197

### 5.3 Authenticator

The authenticator BER-TLV used on the PIV client-application programming interface shall have the structure described in Table 7.

**Table 7 — Data Objects in an Authenticator Template (Tag '67')**

Description	Tag	M/O	Comment
Access mode	'80'	M	
Reference data	'81'	M	
Key reference	'83'	M	See Table 9

### 5.4 Card Application Property Template

The card application property BER-TLV used on the PIV client-application programming interface shall have the structure described in Table 8.

**Table 8 — Data Objects in a Card Application Property Template (Tag '61')**

Description	Tag	M/O	Comment
Application identifier of application	4F'	M	PIX encodes the version of the application
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

### 5.5 Connection Description

**Table 9 — Data Objects in a Connection Description Template**

Description	Tag	M/O	Comment
Interface device identifier – PC/SC	'80'	O	Card reader name
Interface device identifier – SCP	'81'	O	Card reader identifier
Network node identifier – Internet	'84'	O	Internet domain name or IP address
Network node identifier – Telephony	'85'	O	ISDN dialling number string
Exclusive access	'86'	O	If present requests exclusive access to the integrated circuit card

## 5.6 Key References

A key reference is a 6-bit identifier of cryptographic material in the PIV integrated circuit card 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 local (application-specific) reference data.

Table 10 defines the key reference value that shall be used on the PIV interfaces. All other key reference values are reserved for future use.

**Table 10 — PIV Key References**

Key Reference Value	Key Reference Name	Authenticatable Entity	Security Status Type
'00'	Global PIN	Cardholder	Global
'0A'	Card ADM	Card Issuer	Global
'10'	Card Validation Key	Card	Global
'80'	Application PIN	Cardholder	Application
'8A'	Application ADM	Application Provider	Application

## 5.7 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 11. The description of individual client-application programming interface entry points or card commands provide additional information for interpreting the status words they return.

**Table 11 — 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'	'00'	Warning; see entry point or command for specifics
'68'	'00'	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
'69'	'86'	Command not allowed
'6A'	'80'	Incorrect parameter in command data field
'6A'	'81'	Function not supported
'6A'	'84'	Not enough memory

SW1	SW2	Meaning
'6A'	'86'	Incorrect parameter in P1 or P2
'6A'	'88'	Referenced data or reference data not found
'6A'	'89'	Data object already exists
'90'	'00'	Successful execution

## 5.8 Object Identifiers

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 "2.16.840.1.101.3.7.1.1.4.3."

An object identifier on PIV card command interface shall be a BER-TLV data object encoded according to [5]. For example, the representation of the OID of the CHUID on the PIV card command interface is '06 0C 608648016503070101040301'.

## 6. PIV Client-Application Programming Interface

Table 12 lists the entry points on the PIV client-application programming interface.

**Table 12 — Entry Points on PIV Client-Application Programming Interface**

Type	Name
Entry Points for Communication	<b>pivConnect</b>
	<b>pivAcquireContext</b>
	<b>pivReleaseContext</b>
	<b>pivDisconnect</b>
Entry Points for Data Access	<b>pivSelectApplication</b>
	<b>pivGetData</b>
Entry Points for Cryptographic Operations	<b>pivSign</b>
Entry Points for Credential Initialization and Administration	<b>pivPutData</b>
	<b>pivGenerateKeyPair</b>

### 6.1 Entry Points for Communication

#### 6.1.1 pivConnect

**Purpose:** Connects the client-application programming interface and hence the client application itself to a specific PIV integrated circuit card.

**Prototype:**

```

status_word pivConnect (
    IN sequence of bytes      connectionDescription,
    IN boolean                sharedConnection,
    OUT handle                 cardHandle
);

```

**Parameters:** **connectionDescription** Sequence BER-TLV data objects taken from Table 9 above describing a communication path from the platform on which the client-application is running to a specific PIV integrated circuit card.

**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.

**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 pivAcquireContext

**Purpose:** Establishes communication with and a context within a particular card application on the integrated circuit card.

**Prototype:**

```
status_word pivAcquireContext (
    IN handle          cardHandle,
    IN sequence of byte applicationAID,
    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`.

**applicationAID** Identifier of the card application with which the client-application wishes to interact.

**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_APPLICATION_NOT_FOUND
PIV_AUTHENTICATOR_MALFORMED
PIV_AUTHENTICATION_FAILURE

```

### 6.1.3 pivReleaseContext

**Purpose:** Reset the context of the currently selected application including the application security state. The global security state is not affected. There is no currently selected application after successful return from this entry point.

**Prototype:**

```

status_word pivReleaseContext (
    IN handle          cardHandle,
);

```

**Parameters:** **cardHandle** Opaque identifier of the card to be acted upon as returned by pivConnect.

**Return Codes:** PIV\_OK  
PIV\_INVALID\_CARD\_HANDLE  
PIV\_NO\_CURRENT\_CONTEXT

### 6.1.4 pivDisconnect

**Purpose:** Perform a cold reset on the card and disconnect the application programming interface from the PIV and its command interface.

**Prototype:**

```

status_word pivDisconnect (
    IN handle          cardHandle,
);

```

**Parameters:** **cardHandle** Opaque identifier of the card to be acted upon as returned by pivConnect.

**Return Codes:** PIV\_OK  
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 AID             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 properties of the selected card application. See Table 8.

**Return Codes:** PIV\_OK  
PIV\_INVALID\_CARD\_HANDLE  
PIV\_APPLICATION\_NOT\_FOUND

## 6.2.2 pivGetData

**Purpose:** Retrieve data content from the named data object. If data is null then the named data object becomes the currently selected data object in the currently selected application. If length is zero then the entirety of the data content of the data object is returned.

**Prototype:**

```
status_word pivGetData (
    IN handle          cardHandle,
    IN string          OID,
    IN integer         offset,
    IN integer         length,
    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

**offset** The offset into the data content of the data object at which retrieval is to begin

**length** The number of bytes, starting at the offset, to be retrieved from the data object. If length is 0 then the entirety of the data content of the data object is returned.

**data** Array of retrieved data content.

**Return Codes:** PIV\_OK  
PIV\_INVALID\_CARD\_HANDLE  
PIV\_INVALID\_OID  
PIV\_DATA\_OBJECT\_NOT\_FOUND  
PIV\_OFFSET\_BEYOND\_END\_OF\_DATA\_CONTENT  
PIV\_SECURITY\_CONDITIONS\_NOT\_SATISFIED  
PIV\_NO\_CURRENTLY\_SELECTED\_DATA\_OBJECT

## 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             keyReference,
    IN sequence of byte bytesToBeSigned,
    OUT sequence of byte digitalSignature
);
```

**Parameters:**

<b>cardHandle</b>	Opaque identifier of the card to be acted upon as returned by pivConnect.
<b>keyReference</b>	Identifier of reference data to be used to create the digital signature.
<b>bytesToBeSigned</b>	Sequence of bytes, for example a hash, for which a digital signature is to be created.
<b>digitalSignature</b>	The created 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
```

## 6.4 Entry Points for Credential Initialization and Administration

### 6.4.1 pivPutData

**Purpose:** Update the data content of a data object in the currently selected application. If no data is provided and the data object named by the OID is found in the currently selected application then the data object becomes the currently selected data object.

**Prototype:**

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

<b>Parameters:</b>	<b>cardHandle</b>	Opaque identifier of the card to be acted upon as returned by pivConnect.
	<b>OID</b>	Object identifier of the object to be updated
	<b>offset</b>	The offset into the data content of the data object at which updating is to begin.
	<b>length</b>	The number of bytes, starting at the offset, to be written into the data object. If length is zero, then the data in data replaces in its entirety the data content of the data object.
	<b>data</b>	Data to be used to update the data content of the 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

#### 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
);
```

<b>Parameters:</b>	<b>cardHandle</b>	Opaque identifier of the card to be acted upon as returned by Connect.
	<b>keyReference</b>	The key reference of the generated key pair.
	<b>cryptographicMechanism</b>	The type of key pair to be generated. See Table 18

**publicKey**

BER-TLV data objects defining the public key of the generated key pair. See Table 19.

**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

## 7. PIV Card Command Interface

The Table 13 lists the card commands surfaced by the PIV card application at the card edge of the PIV integrated circuit card. All card commands shall be present on conformant and compliant PIV integrated circuit cards.

Table 13 — PIV Card Application Commands

Type	Name	Contact Interface	Contactless Interface
PIV Card Application Commands for Data Access	SELECT APPLICATION	Yes	Yes
	GET DATA	Yes	Yes
PIV Card Application Commands for Authentication	VERIFY PIN	Yes	No
	CHANGE REFERENCE DATA	Yes	No
	RESET RETRY COUNTER	Yes	No
	GENERAL AUTHENTICATE	Yes	See Note
PIV Card Application Commands for Credential Initialization and Administration	PUT DATA	Yes	No
	GENERATE ASYMMETRIC KEY PAIR	Yes	No

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

### 7.1 PIV Card Application Commands for Data Access

#### 7.1.1 SELECT APPLICATION Command

The SELECT APPLICATION 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. The currently selected data object, if any, shall be defined by the selected card application. In the case of PIV card application, the currently selected data object upon selection shall be the CHUID.

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 currently selected data object shall be set to the default data object. The setting of all security status indicators shall be unchanged.

If there is a unique card application the leading bytes of whose application identifier match the provided application identifier, then this card application shall be selected and the complete application identifier of the application returned in the response. As the PIX of the AID contains an encoding of the version of the card application this enables a client-application to select a card application by a generic name and determine the version of the card application that is actually on the PIV integrated circuit card from the returned application property template.

### Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'A4'
<b>P1</b>	'04'
<b>P2</b>	'00' (no response data field) or '0C' (response data field is the selected application's application property template)
<b>L<sub>c</sub></b>	Length of application identifier
<b>Data Field</b>	Application identifier (AID)
<b>L<sub>e</sub></b>	Absent or length of application property template

### Response Syntax

<b>Data Field</b>	Application property template when P2='0C'
<b>SW1-SW2</b>	Status word

<b>SW1</b>	<b>SW2</b>	<b>Meaning</b>
'6A'	'82'	Application not found
'90'	'00'	Successful execution

#### 7.1.2 GET DATA Command

The GET DATA command retrieves the data content of one data object. The number of bytes to be retrieved is given by L<sub>e</sub>.

If an offset is specified in the data field of the command, then the retrieval starts at this byte in the content of the data object. If no offset is provided then all data content is retrieved.

If the OID from which the data is to be retrieved is not specified in the data field of the command then the content of the currently selected data object is returned. Thus, an absent data field returns all data content of the currently selected data object.

If the OID is specified in the data field and if L<sub>e</sub> is equal to zero then the identified data object becomes the currently selected data object.

## Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'CB'
<b>P1</b>	'00'
<b>P2</b>	'00'
<b>L<sub>c</sub></b>	Absent or length of data field
<b>Data Field</b>	See Table 14
<b>L<sub>e</sub></b>	Number of data content bytes to be retrieved.

**Table 14 — Data Objects in the Data Field of the GET DATA Command**

Name	Tag	M/O	Comment
Tag list	'5C'	M	L=0 indicating all Data Objects (DO)
Object identifier	'06'	O	BER-TLV encoding of OID
Offset	'54'	O	Unsigned integer on L bytes

## Response Syntax

<b>Data Field</b>	BER-TLV with the tag '53' or '73' containing in its value field the requested data content. Tag is '73' if the data object named by the OID is a BER-TLV data object and the entire data content of the data object is retrieved
<b>SW1-SW2</b>	Status word

SW1	SW2	Meaning
'63'	'82'	End of data encountered
'69'	'82'	Security status not satisfied
'69'	'85'	Condition of use not satisfied; e.g. no current data object
'6A'	'82'	Data object not found
'90'	'00'	Successful execution

## 7.2 PIV Card Application Commands for Authentication

### 7.2.1 VERIFY PIN Command

The VERIFY PIN 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.

If P2, the key reference, is '00' then the command verifies the cardholder PIN.

## Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'20'
<b>P1</b>	'00'
<b>P2</b>	Key reference
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	Authentication data (i.e., password or PIN)
<b>L<sub>e</sub></b>	Empty

## Response Syntax

<b>SW1</b>	<b>SW2</b>	<b>Meaning</b>
'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 Command

The CHANGE REFERENCE DATA 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.

If P2, the key reference, is '00' then the command applies to the cardholder PIN.

## Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'24'
<b>P1</b>	'00'
<b>P2</b>	Key reference
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	Verification data followed without delimitation by new reference data
<b>L<sub>e</sub></b>	Empty

## Response Syntax

<b>SW1</b>	<b>SW2</b>	<b>Meaning</b>
'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 Command

The RESET RETRY COUNTER 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.

If P2, the key reference, is '00' then the command applies to the cardholder PIN.

#### Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'2C'
<b>P1</b>	'00'
<b>P2</b>	Key reference
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	Verification data followed without delimitation by new reference data
<b>L<sub>e</sub></b>	Empty

#### Response Syntax

<b>SW1</b>	<b>SW2</b>	<b>Meaning</b>
'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 Command

The GENERAL AUTHENTICATE 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 may 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), or efficiently perform a mutual authentication between the card and an entity external to the card (MUTUAL AUTHENTICATE).

The GENERAL AUTHENTICATE command may be used to perform on-card authentication protocols based on biometric data. The GENERAL AUTHENTICATE command may also be used to perform multi-step cryptographic protocols such as those for session key generation, zero knowledge proofs, and protocols involving split and threshold keys.

The GENERAL AUTHENTICATE command is used to realize the signing functionality on the PIV client-application programming interface.

## Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'87'
<b>P1</b>	Algorithm reference
<b>P2</b>	Key reference
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	See Table 15
<b>L<sub>e</sub></b>	Absent or length of expected response

**Table 15 — Data Objects in the Template (Tag '7C')**

Name	Tag	M/O
Witness	'80'	O
Challenge	'81'	O
Response	'82'	O
Committed challenge	'83'	O
Authentication code	'84'	O
Exponential	'85'	O
Identification data template	'A0'	O

## Response Syntax

<b>Data Field</b>	Absent or authentication-related data
<b>SW1-SW2</b>	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 Commands for Credential Initialization and Administration

### 7.3.1 PUT DATA Command

The PUT DATA updates the data content of a data object in the currently selected application.

If no object identifier data object is provided in the data field of the command, then the operation applies to the currently selected data object.

If an offset is specified in the data field of the command, then the update starts at this byte in the content of the data object. If no offset data object is provided in the data field of the command then the data in the data data object in the command data field replaces in entirety the data content of the data object.

## Command Syntax

<b>CLA</b>	'00'
<b>INS</b>	'DB'
<b>P1</b>	'00'
<b>P2</b>	'00'
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	See Table 16
<b>L<sub>e</sub></b>	Empty

**Table 16 — Data Objects in PUT DATA Data Field**

Name	Tag	M/O	Description
Object Identifier	'06'	O	OID of the object being created, updated or deleted
Data	'53' or '73'	O	Data with tag '53' is an unstructured byte sequence. Data with tag '73' is a BER-TLV
Offset	'54'	O	Number of bytes from the start of the content field of the data object where the update is to begin.

## Response Syntax

SW1	SW2	Meaning
'69'	'82'	Security status not satisfied
'69'	'85'	Condition of use not satisfied; e.g. no current data object
'6A'	'84'	Not enough memory
'90'	'00'	Successful execution

### 7.3.2 GENERATE ASYMMETRIC KEY PAIR Command

The GENERATE ASYMMETRIC KEY PAIR 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

<b>CLA</b>	'00'
<b>INS</b>	'47'
<b>P1</b>	'00'
<b>P2</b>	Key reference of generated reference data.
<b>L<sub>c</sub></b>	Length of data field
<b>Data Field</b>	Control reference template. See Table 17.
<b>L<sub>e</sub></b>	Length of public key of data object template

**Table 17 — Data Objects in the Template (Tag 'AC')**

Name	Tag	M/O	Description
Cryptographic mechanism identifier	'80'	M	See Table 18
Parameters	'81'	O	Specific to the cryptographic mechanism

**Table 18 — Cryptographic Mechanism**

Cryptographic Mechanism Identifier	Description
'10'	RSA 1024
'11'	RSA 2048
'20'	ECC 163
'21'	ECC 233
'22'	ECC 283
'23'	ECC 192
'24'	ECC 224
'25'	ECC 256

**Response Syntax**

<b>Data Field</b>	Data objects of public key of generated key pair. See Table 19.
<b>SW1-SW2</b>	Status word

**Table 19 — Data Objects in the Template (Tag '7F49')**

Name	Tag
<b>Public key data objects for RSA</b>	
Modulus	'81'
Public exponent	'82'
<b>Public key data objects for ECDSA</b>	
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

## 8. Use Case Data Flows

The Figures 2 to 5 describe the realization of the four use cases of Section 2 using the high-level client application programming interface described in Section 6 its use of the low-level card command interface described in Section 7.

### 8.1 Card Authentication

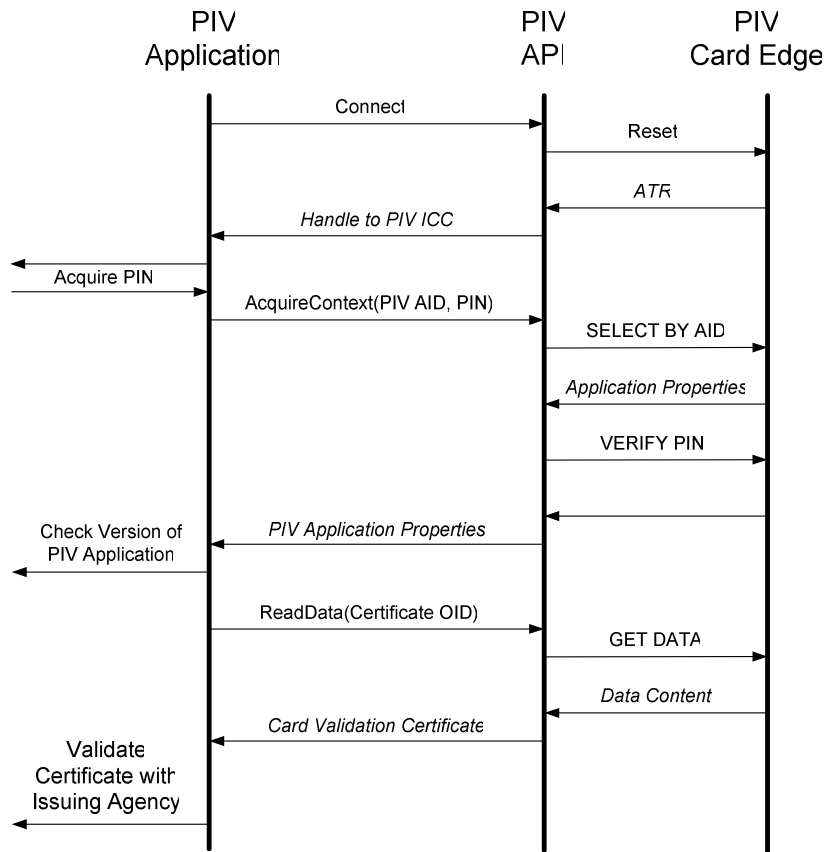


Figure 2 — Data Flow for Card Authentication Use Case

### 8.2 Registration

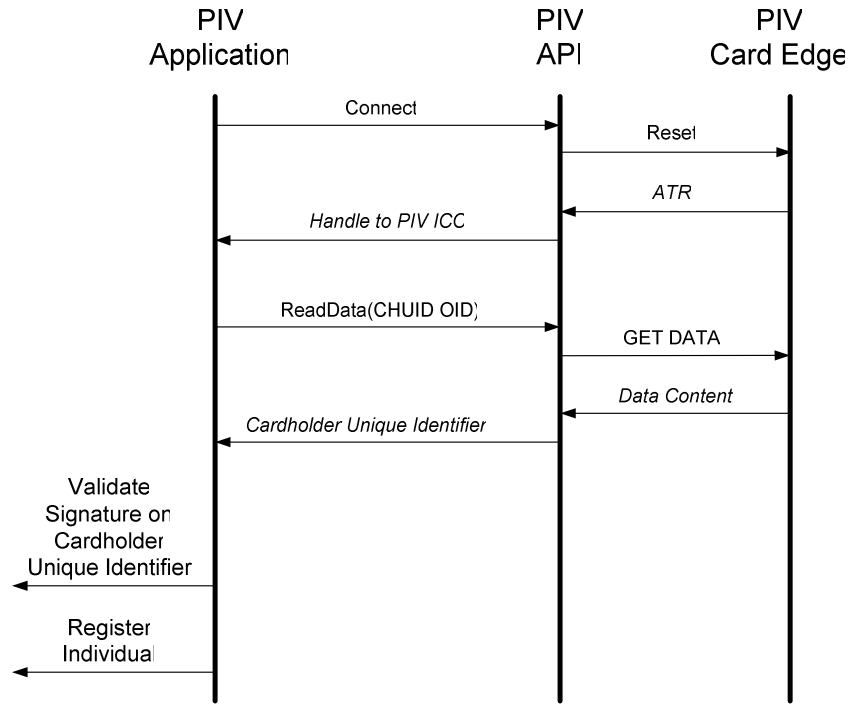


Figure 3 — Data Flow for Registration Use Case

### 8.3 Cardholder Authentication for Physical Access

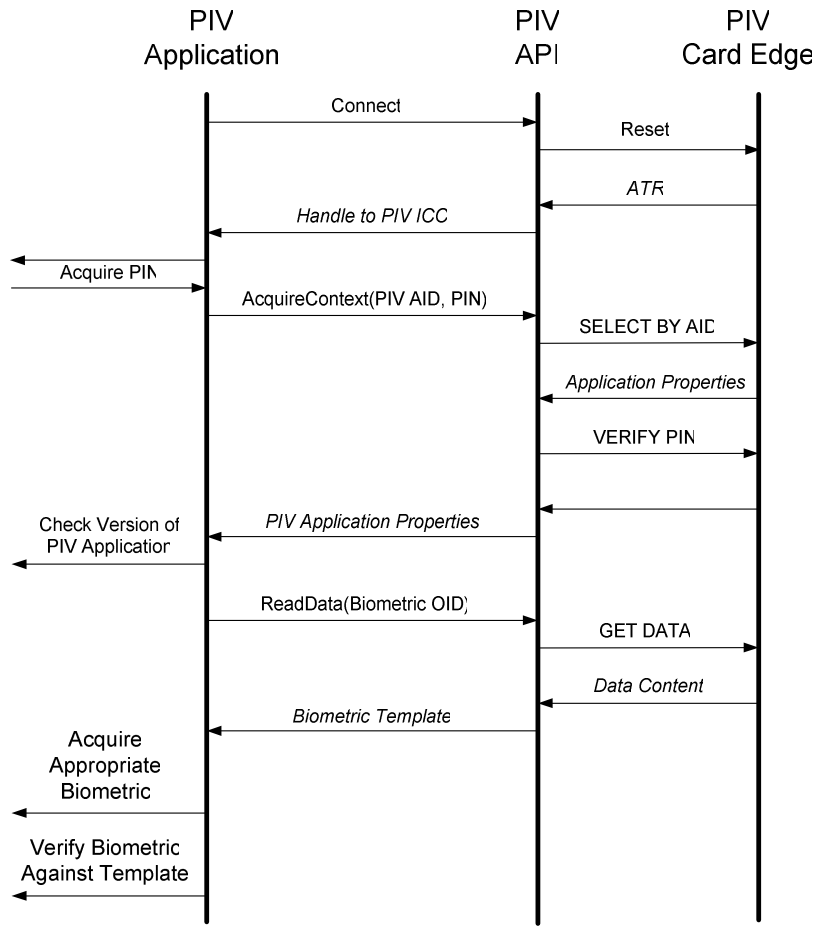


Figure 4 — Data Flow for Cardholder Authentication for Physical Access Use Case

### 8.4 Cardholder Authentication for Logical Access

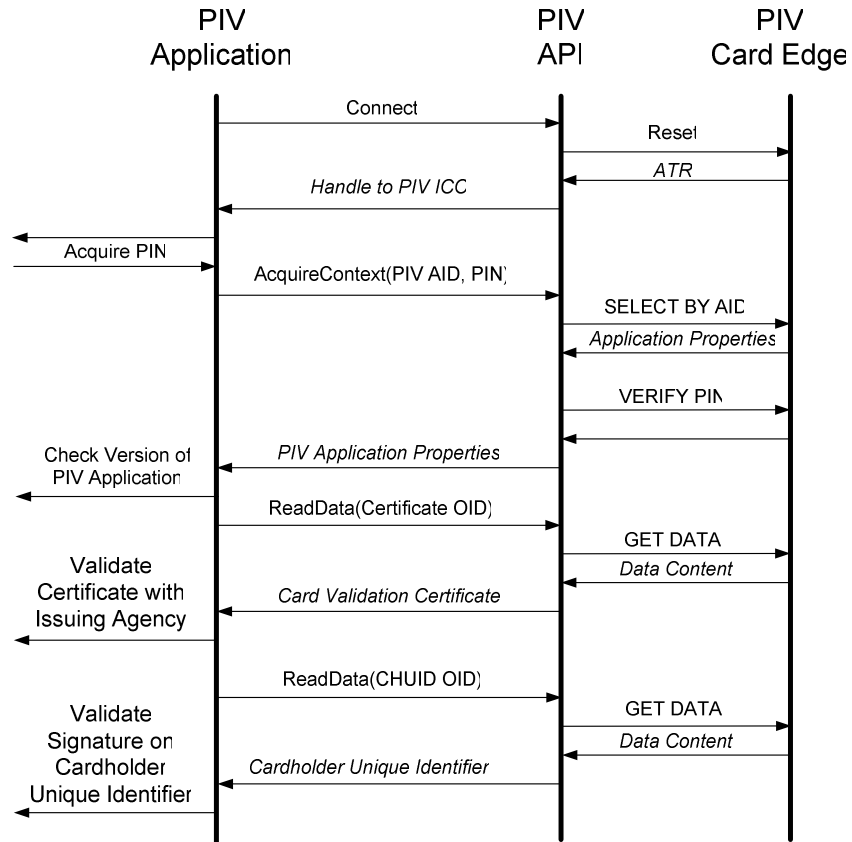


Figure 5 — Data Flow for Cardholder Authentication for Logical Access Use Case

## Appendix A—Examples of GENERAL AUTHENTICATE

Annex C of ISO/IEC 7816-4 [4, Part 4] contains a number of examples of the use of GENERAL AUTHENTICATE including its use to realize the functionality of the INTERNAL AUTHENTICATE, EXTERNAL AUTHENTICATE and MUTUAL AUTHENTICATE card commands.

### A.1 Contactless Card Mutual Authentication

Contactless card style mutual authentication is initiated by the terminal and consists of retrieving a short (e.g. 5-byte) encrypted nonce 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 a 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 20. 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 20 — Mutual Authentication Commands

Terminal Command	Command Data Field	Bytes	Response Data Field	Bytes
GENERAL AUTHENTICATE	'7C' '02' '83' '00'	9	'7C' 'N+2' '80' 'N' {Encryption of card nonce}	N+4+2
GENERAL AUTHENTICATE	'7C' '2N+4' '80' 'N' {Decryption of card nonce} '81' 'N' {Terminal nonce}	11+2N	'7C' 'N+2' '80' 'N' {Encryption of terminal nonce}	N+4+2

## Appendix B—Terms, Acronyms, and Notation

### B.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.
Authenticable 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.
Card Session	The period of time between when a card is reset and either power is removed from the card or the card is again reset.
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.
Reset	An initialization signal sent to an integrated circuit card causing the current state of the card to be set to its defined initialization state. A <i>warm reset</i> is affected by means of a command sent to the card. A <i>cold reset</i> is affected by means of a electrical signal sent to the reset contact of the card processor.
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 specified BER-TLV data objects.

## B.2 Acronyms

AES	Advanced Encryption Standard
AID	Application Identifier
ASCII	American Standard Code for Information Interchange
BER	Basic Encoding Rules
CBC	Circular Binary Coding
CLA	Class (first) byte of a card command
CHUID	Cardholder Unique Identification
DES	Data Encryption Standard
DO	Data Object
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
FASC-N	Federal Agency Smart Credential 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
RSA	Revisi,
SCP	Secure CoPy
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

### **B.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'. Sequences of bytes are enclosed in in apostrophes, for example '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.

In the specification of the PIV integrated circuit card, all bytes specified as reserved for further use (RFU) shall be set to '00' and all bits specified as reserved for further 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 footnotes.

## Appendix C—References

- [1] ISO/IEC 7816 (Parts 4, 5, 6, 8, 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 9564-1:2002, *Banking — Personal Identification Number (PIN) management and security — Part 1: Basic principles and requirements for online PIN handling in ATM and POS systems*.
- [5] 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*
- [6] NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification*, February 2005.
- [7] NIST Federal Information Processing Standards Publication 201, *Personal Identity Verification for Federal Employees and Contractors*, February 2005.
- [8] 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.