NIST Special Publication 800
NIST SP 800-73pt3-5

Interfaces for Personal Identity Verification

Part 3 – PIV Client Application Programming Interface

Hildegard Ferraiolo
Ketan Mehta
Salvatore Francomacaro
Ramaswamy Chandramouli
Sarbari Gupta

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-73pt3-5

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-73pt3-5

Interfaces for Personal Identity Verification

Part 3 – PIV Client Application Programming Interface

Hildegard Ferraiolo
Ketan Mehta
Salvatore Francomacaro
Ramaswamy Chandramouli
Computer Security Division
Information Technology Laboratory

Sarbari Gupta
Electrosoft Services, Inc.

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-73pt3-5

July 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Authority
This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 et seq., Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2024-07-03
Supersedes NIST SP 800-73-4 (May 2015; updated February 8, 2016) https://doi.org/10.6028/NIST.SP.800-73-4

How to Cite this NIST Technical Series Publication:
Ferraiolo H, Mehta K, Francomacaro S, Chandramouli R, Gupta S (2024) Interfaces for Personal Identity Verification: Part 3 – PIV Client Application Programming Interface. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-73pt3-5. https://doi.org/10.6028/NIST.SP.800-73pt3-5

Author ORCID iDs
Hildegard Ferraiolo: 0000-0002-7719-5999
Ketan Mehta: 0009-0001-1191-8656
Salvatore Francomacaro: 0009-0009-0487-2520
Ramaswamy Chandramouli: 0000-0002-7387-5858
Sarbari Gupta: 0000-0003-1101-0856

Contact Information
piv_comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Additional Information
Additional information about this publications is available at https://csrc.nist.gov/pubs/sp/800/73/pt3/5/final including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

FIPS 201 defines the requirements and characteristics of government-wide interoperable identity credentials. It specifies that these identity credentials must be stored on a smart card and that additional common identity credentials, known as derived PIV credentials, may be issued by a federal department or agency and used when a PIV Card is not practical. This document contains the technical specifications to interface with the smart card to retrieve and use the PIV identity credentials. The specifications reflect the design goals of interoperability and PIV Card functions. The goals are addressed by specifying a PIV data model, card edge interface, and application programming interface. Moreover, this document enumerates requirements for the options and branches in international integrated circuit card standards. The specifications go further by constraining interpretations of the normative standards to ease implementation, facilitate interoperability, and ensure performance in a manner tailored for PIV applications.

Keywords

authentication; FIPS 201; identity credential; logical access control; on-card biometric comparison; Personal Identity Verification (PIV); physical access control; smart cards; secure messaging.

Reports on Computer Systems Technology

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication. As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL. No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

List of Tables

Acknowledgments

The authors — Hildegard Ferraiolo, Ketan Mehta, Salvatore Francomacaro, and Ramaswamy Chandramouli of NIST and Sarbari Gupta of Electrosoft Services, Inc. — gratefully acknowledge the contributions of David Cooper, James Dray, William MacGregor, Scott Guthery, Teresa Schwarzhoff, and Jason Mohler, who co-authored prior versions of this three-part publication. 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.

1. Introduction

Homeland Security Presidential Directive-12 (HSPD-12) called for the adoption of a common identification standard to govern the interoperable use of identity credentials to allow physical and logical access to federally controlled facilities and information systems. In response, Federal Information Processing Standard (FIPS) 201 [FIPS201], *Personal Identity Verification (PIV) of Federal Employees and Contractors*, was developed to define reliable, government-wide identity credentials for use in applications such as access to federally controlled facilities and information systems. FIPS 201 supports multiple types of authenticators, including authenticators on smart cards (also known as PIV Cards) and derived PIV credential authenticators in various other form factors. This publication contains technical specifications to interface with PIV Cards to retrieve and use identity credentials. Other specifications, such as NIST Special Publication (SP) 800-157r1 (Revision 1), contain procedures and life cycle activities to issue, maintain, and use derived PIV credentials.

1.1. Purpose

FIPS 201 defines processes for binding identities to authenticators, such as the PIV Card and derived PIV credentials used in the federal PIV system. SP 800-73-5 contains the technical specifications to interface with the PIV Card to retrieve and use the identity credentials. The specifications reflect the design goals of interoperability and PIV Card functions. The goals are addressed by specifying a PIV data model, card edge interface, and application programming interface. Moreover, this document enumerates requirements for the options and branches in international integrated circuit card (ICC) standards [ISO7816]. The specifications go further by constraining interpretations of the normative standards to ease implementation, facilitate interoperability, and ensure performance in a manner tailored for PIV applications.

1.2. Scope

SP 800-73-5 specifies the PIV data model, application programming interface (API), and card interface requirements necessary to comply with the use cases, as defined in Section 6 of FIPS 201 and further described in Appendix B of SP 800-73-5 Part 1. Interoperability is defined as the use of PIV identity credentials such that client-application programs, compliant card applications, and compliant ICCs CAN be used interchangeably by all information processing systems across federal agencies. SP 800-73-5 defines the PIV data elements' identifiers, structure, and format, as well as the client API and card command interface for use with the PIV Card.

This document — SP 800-73-5, *Interfaces for Personal Identity Verification: Part 3 – PIV Client Application Programming Interface* — contains technical specifications for the PIV client application programming interface to the PIV Card.

1.3. Audience and Assumptions

This document is intended for federal agencies and implementers of PIV systems. Readers are assumed to have a working knowledge of smart card standards and applications.

Readers should also be aware of the following important content in SP 800-73-5 Part 1:

- The front matter details configuration management recommendations and specifies NPIVP conformance testing procedures.
- Appendix G provides the full Revision History of SP 800-73.
- Section 1.3 specifies the effective date of SP 800-73-5.

1.4. Content and Organization

All sections in this document are *normative* (i.e., mandatory for compliance) unless specified as *informative* (i.e., non-mandatory) and are structured as follows:

- Section 1, *Introduction*, provides the purpose, scope, audience, and assumptions of the document and outlines its structure.
- Section 2*, Overview: Concepts and Constructs,* describes both the PIV Card Application and the PIV client API. This section is *informative*.
- Section 3, *Client Application Programming Interface,* describes the set of entry points accessible by client applications through the PIV Middleware to interact with the PIV Card.
- Appendix A contains the list of acronyms used in this document. This section is *informative*.
- Appendix B contains the Glossary of terms used in this document. This section is *informative*.
- Appendix C explains the notation in use in this document. This section is *informative*.

2. Overview: Concepts and Constructs

SP 800-73-5 Parts 2 and 3 define two interfaces to an ICC that contain the PIV Card Application: a low-level card command interface (SP 800-73-5 Part2) and a high-level client API (Part 3). SP 800-73-5 Part 3 (this document) is optional, and NIST Personal Identity Verification Program (NPIVP) conformance testing for PIV Middleware in accordance with SP 800-73 Part 3 is discontinued since endpoints support high-level client APIs natively at the time of this publication.

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 API or the card command interface.

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

The client API is thought of as being at a higher level than the card command interface because access to a single entry point on the client API 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 of the client API.

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

3. Client Application Programming Interface

Table 1 lists the entry points on the PIV client API. This section references object identifiers (OIDs), which are defined and can be found in SP 800-73-5 Part1, Table 3.

Table 1. Entry points on PIV client application programming interface

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

If both the PIV Middleware and the PIV Card support secure messaging, then all non-card management functionality[1] of the PIV Card MAY be accessed over either the contact or contactless interface of the card. In order to perform non-card management functionality that would otherwise be limited to the contact interface, the client application must first establish a virtual contact interface by calling the pivEstablishSecureMessaging function and using the pivLogIntoCardApplication function to submit the pairing code to the card.[2] If the client application does not have another means of determining whether communication with the PIV Card is over a contact or contactless interface, it MAY use the pivGetData function to attempt to read a mandatory data object (e.g., such as the X.509 Certificate for PIV Authentication or the security object) that has an access rule for read of "Always." However, that is only accessible over the contact and virtual contact interfaces (see SP 800-73-5 Part1, Table 2). If the return code from pivGetData is `PIV_SECURITY_CONDITIONS_NOT_SATISFIED`, then the communication with the card is over a contactless interface.

[1] Only the pivPutData and pivGenerateKeyPair API functions perform card management functionality.

[2] As noted in Part 1, Sec. 5.5, the pairing code does not need to be submitted if the Bit 3 of the first byte of the PIN Usage Policy is set to one.

3.1. Entry Points for Communication

3.1.1. pivMiddlewareVersion

Purpose: Returns the PIV Middleware version string

Prototype:

```
status_word pivMiddlewareVersion(
   OUT   version           versionString
);
```

Parameter: **versionString**

- For SP 800-73-5 Part 3 conformant PIV Middleware, the parameter returns “800-73-5 Client API” or “800-73-5 Client API with SM.”
- For SP 800-73-4 Part 3 conformant middleware, the parameter returns “800-73-4 Client API” or “800-73-4 Client API with SM.”
- Earlier versions with versionString (e.g., “800-73-3 Client API,” “800-73-2 Client API,” and “800-78-1 Client API”) are discontinued as they do not conform to the requirements of FIPS 201-3.

Return Codes: `PIV_OK`

PIV Middleware that returns a versionString of “800-73-5 Client API with SM” and “800-73-4 Client API with SM” SHALL implement all PIV Middleware functions listed in **Table 1** and be able to recognize and process all mandatory and optional PIV data objects. PIV Middleware that returns a versionString of “800-73-5 Client API” and “800-73-4 Client API” SHALL implement all PIV Middleware functions listed in **Table 1** except for pivEstablishSecureMessaging and SHALL be able to recognize and process all mandatory and optional PIV data objects.

3.1.2. pivConnect

Purpose: Connects the client API to the PIV Card Application on a specific ICC.

Prototype:

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

Parameters:

sharedConnection If TRUE, other client applications CAN stablish concurrent connections to the ICC. If FALSE and the connection is established, then the calling client application has exclusive access to the ICC.

connectionDescription A connection description data object (tag 0x7F21). See **Table 2**.

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.

In order to provide sufficient space for the return value, the client application SHALL allocate a buffer of at least 2048 bytes for connectionDescription.

The connection description BER-TLV [ISO8825] used on the PIV client API SHALL have the structure described in **Table 2**.

Table 2. Data objects in a connection description template (Tag 0x7F21)

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

At most one selection from the ‘8x’ series and one selection from the ‘9x’ series SHALL appear in the connection description template. For example, ‘7F 21 0C 82 04 41 63 6D 65 91 04 C0 00 02 17’ describes a connection to a generic card reader at internet address 192.0.2.23. In another example, ‘7F 21 0B 82 01 00 93 06 16 17 55 50 12 3F’ describes a connection to the subscriber identity module in the mobile phone at +1 617 555 0123.

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

CDLength Length of the card description parameter

cardHandle The returned opaque identifier of a communication channel to a particular ICC and, hence, of the card itself. cardHandle is used in all other entry points on the PIV client API to identify which card the functionality of the entry point should be applied to.

Return Codes:

```
PIV_OK
PIV_CONNECTION_DESCRIPTION_MALFORMED
PIV_CONNECTION_FAILURE
PIV_CONNECTION_LOCKED
```

3.1.3. pivDisconnect

Purpose: Disconnect the PIV API from the PIV Card Application and the ICC that contains the PIV Card Application.

Prototype:

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

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

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_READER_ERROR
```

If secure messaging has been established, then the PIV Middleware SHALL zeroize the secure messaging session keys.

3.2. Entry Points for Data Access

3.2.1. pivSelectCardApplication

Purpose: Set the PIV Card Application as the currently selected card application and establish the PIV Card Application’s security state.

Prototype:

```
status_word pivSelectCardApplication(
IN handle                cardHandle,
IN sequence of byte      applicationAID,
IN LONG                  aidLength,
OUT sequence of byte     applicationProperties,
INOUT LONG              APLength
);
```

Parameters:

cardHandle Opaque identifier of the card to be acted upon as returned by pivConnect

aidLength Length of the PIV Card Application AID

applicationAID The AID of the PIV Card Application that is to become the currently selected card application

applicationProperties The application properties of the selected PIV Card Application; see SP 800-73-5 Part2, Table 3

APLength As an input, length of the buffer allocated for applicationProperties; as an output, length of the application properties

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_CARD_APPLICATION_NOT_FOUND
PIV_CARD_READER_ERROR
PIV_INSUFFICIENT_BUFFER
```

If the length of application properties is longer than the buffer allocated by the client application, then the PIV Middleware SHALL return `PIV_INSUFFICIENT_BUFFER` but SHALL still set APLength to the length of the application properties.

3.2.2. pivEstablishSecureMessaging

Purpose: Establish secure messaging with the PIV Card Application.

Prototype:

```
status_word pivEstablishSecureMessaging(
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_CARD_READER_ERROR
PIV_SM_FAILED
```

After successful execution of the key establishment protocol, the PIV Middleware SHALL perform all subsequent GET DATA, VERIFY, and GENERAL AUTHENTICATE commands over secure messaging with the exception of any subsequent uses of the GENERAL AUTHENTICATE command to perform the key establishment protocol.

3.2.3. pivLogIntoCardApplication

Purpose: Set the security state within the PIV Card Application.

Prototype:

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

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 and set the security state/status in the PIV Card Application context. The authenticator BER-TLV used on the PIV client API SHALL have the structure described in **Table 3**.
AuthLength	Length of the authenticator template.

Table 3. Data objects in an authenticator template (Tag '67')

Description	Tag	M/O	Comment
Reference data	'81'	M	Value of the PIV Card Application PIN, Global PIN, or pairing code, as described in Sec. 2.4.3 of SP 800-73-5 Part2, or OCC data, as described in Sec. 5.5.2 of [SP800-76]
Key reference	'83'	M	See Table 4 of SP 800-73-5 Part1 for PIV Card Application PIN, Global PIN, pairing code, and OCC key reference values

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_AUTHENTICATOR_MALFORMED
PIV_AUTHENTICATION_FAILURE
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_CARD_READER_ERROR
PIV_SM_FAILED
```

The PIV Middleware SHALL NOT submit authenticators to the PIV Card over a contactless interface without secure messaging. If secure messaging has not been established, then the pivLogIntoCardApplication function SHALL return `PIV_SECURITY_CONDITIONS_NOT_SATISFIED`.

3.2.3.1. pivGetData

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

Prototype:

```
status_word pivGetData(
      IN handle                cardHandle,
      IN string                OID,
      IN LONG                  oidLength,
      OUT sequence of byte     data,
      INOUT LONG               DataLength
      );
```

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 (e.g., "2.16.840.1.101.3.7.2.96.80"). See SP 800-73-5 Part1, Table 3.
oidLength	Length of the object identifier.
data	Retrieved data content.
DataLength	As an input, length of the buffer allocated for data. As an output, length of the data retrieved from the PIV Card.

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_INVALID_OID
PIV_DATA_OBJECT_NOT_FOUND
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_CARD_READER_ERROR
PIV_SM_FAILED
PIV_INSUFFICIENT_BUFFER
```

If the length of the retrieved data is longer than the buffer allocated by the client application, then the PIV Middleware SHALL return PIV_INSUFFICIENT_BUFFER but SHALL still set DataLength to the length of the retrieved data. If the PIV Card Application returns a zero-length data object, then the PIV Middleware SHALL return PIV_DATA_OBJECT_NOT_FOUND.

3.2.4. pivLogoutOfCardApplication

Purpose: Reset the application security state/status of the PIV Card Application.

Prototype:

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

Parameters:

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

Return Codes:

```
PIV_OK
        PIV_INVALID_CARD_HANDLE
        PIV_CARD_READER_ERROR
```

3.3. Entry Points for Cryptographic Operations

3.3.1. pivCrypt

Purpose: Perform a cryptographic operation,[3] such as encryption or signing on a sequence of bytes. SP 800-73-5 Part1, Appendix C describes recommended procedures for PIV algorithm identifier discovery.

Prototype:

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

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect
algorithmIdentifier	Identifier of the cryptographic algorithm to be used for the cryptographic operation [SP800-78, Tables 9 and 10]
keyReference	Identifier of the on-card key to be used for the cryptographic operation. See [SP800-78, Table 8] and SP 800-73-5 Part1, Table 5.

[3] The pivCrypt function does not perform any cryptographic operations itself. It provides the interface to the GENERAL AUTHENTICATE command to perform cryptographic operations on-card. All cryptographic operations are performed outside of the PIV Middleware except for SM on the client side.

`algorithmInput` Sequence of bytes used as the input to the cryptographic operation. The algorithmInput for RSA algorithms SHALL be restricted to the range 0 to *n*-1, where *n* is the RSA modulus.

`inputLength` Length of the algorithm input

`algorithmOutput` Sequence of bytes output by the cryptographic operation

`outputLength` As an input, length of the buffer allocated for algorithmOutput. As an output, length of the algorithm output.

Return Codes: `PIV_OK`

`PIV_INVALID_CARD_HANDLE`

`PIV_INVALID_KEYREF_OR_ALGORITHM`

`PIV_SECURITY_CONDITIONS_NOT_SATISFIED`

`PIV_INPUT_BYTES_MALFORMED`

`PIV_CARD_READER_ERROR`

`PIV_SM_FAILED`

`PIV_INSUFFICIENT_BUFFER`

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

If the value of keyReference is '04' (PIV Secure Messaging key), then the PIV Middleware SHALL return `PIV_INVALID_KEYREF_OR_ALGORITHM`.

If the length of the algorithm output is longer than the buffer allocated by the client application, then the PIV Middleware SHALL return `PIV_INSUFFICIENT_BUFFER` but SHALL still set outputLength to the length of the algorithm output.

3.4. Entry Points for Credential Initialization and Administration

The PIV Middleware SHALL NOT submit data provided to the pivPutData or pivGenerateKeyPair function over the contactless interface. If the PIV Middleware is not communicating with the PIV Card via the card's contact interface, then the pivPutData or pivGenerateKeyPair function SHALL return `PIV_FUNCTION_NOT_SUPPORTED`.

3.4.1. 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 LONG                 oidLength,
IN sequence of byte     data,
IN LONG                 dataLength
);
```

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 (e.g., “2.16.840.1.101.3.7.2.96.80”). See SP 800-73-5 Part1, Table 3.

oidLength Length of the object identifier

data Data to be used to replace in its entirety the data content of the named data object

dataLength Length of the provided data

Return Codes:

```
PIV_OK
PIV_INVALID_CARD_HANDLE PIV_INVALID_OID PIV_CARD_READER_ERROR
PIV_INSUFFICIENT_CARD_RESOURCE
PIV_SECURITY_CONDITIONS_NOT_SATISFIED
PIV_FUNCTION_NOT_SUPPORTED
```

3.4.2. pivGenerateKeyPair

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

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

Prototype:

```
status_word pivGenerateKeyPair(
IN handle                       cardHandle,
IN byte                         keyReference,
IN byte                         cryptographicMechanism,
OUT sequence of byte            publicKey,
INOUT LONG                      KeyLength
);
```

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 SP 800-73-5 Part1, Table 7.

publicKey BER-TLV data objects defining the public key of the generated key pair. See SP 800-73-5 Part2, Table 11.

KeyLength As an input, the length of the buffer allocated for publicKey; as an output, length of the public key-related data retrieved from the PIV Card

Return Codes: PIV_OK

PIV_INVALID_CARD_HANDLE

PIV_SECURITY_CONDITIONS_NOT_SATISFIED

PIV_FUNCTION_NOT_SUPPORTED

PIV_INVALID_KEY_OR_KEYALG_COMBINATION

PIV_UNSUPPORTED_CRYPTOGRAPHIC_MECHANISM

PIV_CARD_READER_ERROR

PIV_INSUFFICIENT_BUFFER

If the length of public key-related data retrieved from the PIV Card is longer than the buffer allocated by the client application, then the PIV Middleware SHALL return PIV_INSUFFICIENT_BUFFER but SHALL still set KeyLength to the length of the public key-related data retrieved from the PIV Card.

References

[FIPS201] National Institute of Standards and Technology (2022) Personal Identity Verification (PIV) of Federal Employees and Contractors. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) 201-3. https://doi.org/10.6028/NIST.FIPS.201-3

[ISO7816] International Organization for Standardization/International Electrotechnical Commission (2004-2020) ISO/IEC 7816 — Identification cards — Integrated circuit cards. (multiple parts):

- International Organization for Standardization/International Electrotechnical Commission (2020) ISO/IEC 7816-4:2020 — Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange. (International Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/77180.html
- International Organization for Standardization/International Electrotechnical Commission (2004) ISO/IEC 7816-5:2004 — Identification cards — Integrated circuit cards — Part 5: Registration of application providers. (International Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/34259.html
- International Organization for Standardization/International Electrotechnical Commission (2023) ISO/IEC 7816-6:2023 — Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange. (International Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/77181.html
- International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 7816-8:2021 — Identification cards — Integrated circuit cards — Part 8: Commands and mechanisms for security operations. (International Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/79893.html
- International Organization for Standardization/International Electrotechnical Commission (2017) ISO/IEC 7816-9:2017 — Identification cards — Integrated circuit cards — Part 9: Commands for card management. (International Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/67802.html

[ISO8825] International Organization for Standardization/International Electrotechnical Commission (2015) ISO/IEC 8825-1:2015— Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) Part 1. (International

Organization for Standardization, Geneva, Switzerland) [or as amended]. Available at https://www.iso.org/standard/81420.html

[SP800-76] Grother PJ, Salamon WJ, Chandramouli R (2013) Biometric Specifications for Personal Identity Verification. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-76-2 [or as amended]. https://doi.org/10.6028/NIST.SP.800-76-2

[SP800-78] Ferraiolo H, Regenscheid A (2024) Cryptographic Algorithms and Key Sizes for Personal Identity Verification. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-78-5 [or as amended]. https://doi.org/10.6028/NIST.SP.800-78-5

Appendix A. List of Symbols, Abbreviations, and Acronyms

AID
Application Identifier

API
Application Programming Interface

ASN.1
Abstract Syntax Notation One

BER
Basic Encoding Rules

FIPS
Federal Information Processing Standard

FISMA
Federal Information Security Management Act

GSC-IS
Government Smart Card Interoperability Specification

HSPD
Homeland Security Presidential Directive

ICC
Integrated Circuit Card

IEC
International Electrotechnical Commission

INCITS
InterNational Committee for Information Technology Standards

ISDN
Integrated Services Digital Network

ISO
International Organization for Standardization

ITL
Information Technology Laboratory

LSB
Least Significant Bit

MSB
Most Significant Bit

NIST
National Institute of Standards and Technology

OCC
On-Card Biometric Comparison

OID
Object Identifier

OMB
Office of Management and Budget

PC/SC
Personal Computer/Smart Card

PIN
Personal Identification Number

PIV
Personal Identity Verification

PKCS
Public-Key Cryptography Standards

PKI
Public Key Infrastructure

RFU
Reserved for Future Use

SM
Secure Messaging

SP
Special Publication

TLV
Tag-Length-Value

Appendix B. Glossary

application identifier
A globally unique identifier of a card application, , as adapted from 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.

algorithm identifier
A 1-byte identifier that specifies a cryptographic algorithm and key size. For symmetric cryptographic operations, the algorithm identifier also specifies a mode of operation (i.e., ECB).

BER-TLV data object
A data object coded according to ISO/IEC 8824-2:2021.

card
An integrated circuit card.

card application
A set of data objects and card commands that can be selected using an application identifier.

card interface device
An electronic device that connects an integrated circuit card and the card applications therein to a client application.

card reader
Synonym for *card interface device*.

client application
A computer program running on a computer in communication with a card interface device.

card management operation
Any operation involving the PIV Card Application Administrator.

data object
An item of information seen at the card command interface for which is specified a name, a description of logical content, a format, and a coding.

interface device
Synonym for *card interface device*.

key reference
A 1-byte identifier that specifies a cryptographic key according to its PIV Key Type. The identifier is part 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 adapted from ISO/IEC 8824-2:2021.

reference data
Cryptographic material used in the performance of a cryptographic protocol, such as an authentication or a signing protocol. The reference data length is the maximum length of a password or PIN. For algorithms, the reference data length is the length of a key.

status word
Two bytes returned by an integrated circuit card after processing any command that encodes 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.

Appendix C. Notation

The 16 hexadecimal digits SHALL be denoted using the alphanumeric characters 0, 1, 2, …, 9, A, B, C, D, E, and F. A byte consists of two hexadecimal digits, for example, '2D'. The two hexadecimal digits are represented in quotations '2D' or as 0x2D. A sequence of bytes MAY be enclosed in single quotation marks (e.g., 'A0 00 00 01 16') rather than given as a sequence of individual bytes (e.g., 'A0' '00' '00' '01' '16').

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

All bytes specified as RFU SHALL be set to '00', and all bits specified as RFU 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 that the data object SHALL appear in the template. Optional means that the data object MAY appear in the template.

In other tables the M/O/C column identifies properties of the PIV Card Application that SHALL be present (M), may be present (O), or are conditionally required to be present (C).

BER-TLV data object tags are represented as byte sequences, as described above. Thus, for example, 0x4F is the interindustry data object tag for an application identifier, and 0x7F60 is the interindustry data object tag for the Biometric Information Templates Group template.

This document uses the following typographical conventions in text:

- ASN.1 data types are represented in a monospaced font. For example, SignedData and SignerInfo are data types defined for digital signatures.
- Specific terms in **CAPITALS** represent normative requirements. When these same terms are not in **CAPITALS**, the term does not represent a normative requirement.
- The terms **SHALL** and **SHALL NOT** indicate requirements to be followed strictly in order to conform to the publication and from which no deviation is permitted.
- The terms **SHOULD** and **SHOULD NOT** indicate that among several possibilities, one is recommended as particularly suitable without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that — in the negative form — a certain possibility or course of action is discouraged but not prohibited.
- The terms **MAY** and **NEED NOT** indicate a course of action that is permissible within the limits of the publication.
- The terms **CAN** and **CANNOT** indicate a material, physical, or causal possibility or capability or — in the negative — the absence of that possibility or capability.