Archived NIST Technical Series Publication

The attached publication has been archived (withdrawn), and is provided solely for historical purposes. It may have been superseded by another publication (indicated below).

Archived Publication

Series/Number:	NIST Special Publication 800-73-3
Title:	Interfaces for Personal Identity Verification
Publication Date(s):	February 2010
Withdrawal Date:	May 2015
Withdrawal Note:	SP 800-73-3 is superseded in its entirety by the publication of SP 800-73-4 (May 2015).

Superseding Publication(s)

The attached publication has been **superseded by** the following publication(s):

Series/Number:	NIST Special Publication 800-73-4
Title:	Interfaces for Personal Identity Verification
Author(s):	David Cooper, Hildegard Ferraiolo, Ketan Mehta, Salvatore Francomacaro, Ramaswamy Chandramouli, Jason Mohler
Publication Date(s):	May 2015
URL/DOI:	http://dx.doi.org/10.6028/NIST.SP.800-73-4

Additional Information (if applicable)

Contact:	Computer Security Division (Information Technology Lab)
Latest revision of the attached publication:	SP 800-73-4 (as of August 6, 2015)
Related information:	http://csrc.nist.gov/groups/SNS/piv/
Withdrawal announcement (link):	N/A

Date updated: August 6, 2015

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

NIST Special Publication 800-73-3

Interfaces for Personal Identity Verification – Part 1: End-Point PIV Card Application Namespace, Data Model and Representation

Ramaswamy Chandramouli
David Cooper
James F. Dray
Hildegard Ferraiolo
Scott B. Guthery
William MacGregor
Ketan Mehta

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

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

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

February 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

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

National Institute of Standards and Technology Special Publication 800-73-3, Part 1, 56 pages, February 2010)

Acknowledgements

The authors (Ramaswamy Chandramouli, David Cooper, James Dray, Hildegard Ferraiolo, William MacGregor of NIST, Ketan Mehta of Booz Allen Hamilton, and Scott Guthery of HID Global) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Government Smart Card Interagency Advisory Board (GSC-IAB) and InterNational Committee for Information Technology Standards (INCITS) for providing detailed technical inputs to the SP 800-73 development process. The authors also gratefully acknowledge and appreciate the many contributions from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

I. Revision History

Version	Release Date	Updates
SP 800-73	April 2005	Initial Release
SP 800-73-1	April 2006	Incorporated Errata
SP 800-73-2	September 2008	• Separated SP 800-73 into four Parts: 1 - *End-Point PIV Card Application Namespace, Data Model and Representation* *2 - End-Point PIV Card Application Card Command Interface* *3 - End-Point PIV Client Application Programming Interface* *4 - The PIV Transitional Interface and Data Model Specification* • All PIV cryptographic key types, cryptographic algorithm identifiers, and key sizes previously listed in SP 800-73-1, are now specified in SP 800-78, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification* • Removed default algorithms. Each PIV key type can be implemented from a small subset of algorithms and key sizes as specified in Table 3-1 of SP 800-78 • Added optional Discovery Object (Part 1, Section 3.2.6) • Added optional capability to use the Global PIN (in addition to the PIV Card Application PIN) with the PIV Card Application (Part 1, Section 3.2.6) • Added pivMiddlewareVersion API function (Part 3, Section 3.1.1) • Deprecated the CHUID Data Object's Authentication Key Map data element • Deprecated the Printed Information Data Object's Employee Affiliation Line 2 (tag 0x03) • Removed size limits on signed data object containers (Part 1, Appendix A)
SP 800-73-3	February 2010	• Added preamble: I - Revision History, II - Configuration Management and III – NPIVP Conformance Testing. (Part 1, Preamble) • Removed the CHUID's Authentication Key Map data element • Removed the Printed Information's Employee Affiliation Line 2 data element (tag 0x03) • Deprecated IPv6 as optional value for the CHUID's GUID data element (Part 1, Section 3.2.1) • Added Key History capability (Part 1, Section 3.2.7) • Added ECDH key agreement scheme (Part 2, Section 3.2.4) • Added UUID feature for NFI cards (Part 1, Section 3.3) • Expanded Part 2, Appendix A (GENERAL AUTHENTICATE examples) to illustrate ECDSA signatures and key establishment schemes with the Key Management Key • Added an optional Cardholder Iris Images Data Object, which will be specified in SP 800-76-2.

The Revision History is a list of updates to SP 800-73 since its initial release. All updates are optional additions to the initial release of SP 800-73. Therefore, current PIV Cards with or without these optional features remain valid.

Deprecated or removed items in the revision history do not affect current PIV Cards in circulation. PIV Cards with deprecated/removed data elements remain valid until they naturally expire. Replacement PIV Cards, however, should not re-use the deprecated/removed data elements.

II. Configuration Management

When a Federal Agency adds one or several optional features listed in the previous section (Revision History) to their PIV Cards, it is necessary for client applications to upgrade the PIV Middleware accordingly. This will enable the PIV Middleware to recognize and process the new data objects and/or features.

Where maximum interoperability is required, it is necessary to upgrade to SP 800-73-3 based PIV Middleware as they become available. Only SP 800-73-3 based PIV Middleware fully support all capabilities outlined in the Revision History. Previous versions of the PIV Middleware (based on SP 800-73-2 or SP 800-73-1) are unaware of SP 800-73-3 features and thus have the following limitations:

+ SP 800-73-1 based PIV Middleware:
    - Do not recognize PIV Discovery Objects and thus are unable to recognize or prompt for the Global PIN for PIV Cards with Global PIN enabled.
    - Do not recognize the Key History feature.

    Recommendation: SP 800-73-1 based PIV Middleware should be restricted to applications that do not use any of the optional features outlined in the Revision History in Section I.

+ SP 800-73-2 based PIV Middleware:
    - Recognize the Global PIN of PIV Cards with Global PIN enabled, but
    - Do not support the Key History feature.

    Recommendation: SP 800-73-2 based PIV Middleware should be restricted to applications that do not use any SP 800-73-3 based optional features outlined in the Revision History in Section I.

III NPIVP Conformance Testing

As outlined in FIPS 201-1, Appendix B-3, NIST has established the NIST Personal Identity Verification Program (NPIVP) to:

+ validate the compliance/conformance of two PIV components: PIV Middleware and PIV Card Applications with the specifications in NIST SP 800-73 and
+ provide the assurance that the set of PIV Middleware and PIV Card Applications that have been validated by NPIVP are interoperable.

For the further information on NPIVP, see http://csrc.nist.gov/groups/SNS/piv/npivp/index.html.

With the final release of SP 800-73-3, NPIVP plans to revise and publish SP 800-85A-2, PIV Card Application and Middleware Interface Test Guidelines. This document will outline the Derived Test Requirements (DTRs) of SP 800-73-3 based PIV Card Applications and PIV Middleware. In parallel, NPIVP plans to update the test tools for NPIVP laboratories to test PIV Card Applications and PIV Middleware in accordance with the DTRs in SP 800-85A-2. Once SP 800-85A-2 is published, and the test tools are available to NPIVP test laboratories, SP 800-73-2 based testing will be discontinued and SP 800-73-3 based testing will begin. NPIVP will announce the start of SP 800-73-3 based testing at http://csrc.nist.gov/groups/SNS/piv/npivp/announcements.html.

TABLE OF CONTENTS

List of Appendices

List of Tables

List of Figures

1. Introduction

The Homeland Security Presidential Directive 12 (HSPD-12) called for a common identification standard to be adopted governing the interoperable use of identity credentials to allow physical and logical access to Federal government locations and systems. The Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201 (FIPS 201) [1] was developed to establish standards for identity credentials. Special Publication 800-73-3 (SP 800-73-3) contains technical specifications to interface with the smart card (PIV Card[1]) to retrieve and use the identity credentials.

1.1 Authority

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

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

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

1.2 Purpose

FIPS 201 defines procedures for the PIV lifecycle activities including identity proofing, registration, PIV Card issuance, and PIV Card usage. FIPS 201 also specifies that the identity credentials must be stored on a smart card. SP 800-73-3 contains the technical specifications to interface with the smart 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, SP 800-73-3 enumerates requirements where the standards include options and branches. The specifications go further by constraining implementers' interpretations of the normative standards. Such restrictions are designed to ease implementation, facilitate interoperability, and ensure performance, in a manner tailored for PIV applications.

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

1.3 Scope

SP 800-73-3 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 this document. Interoperability is defined as the use of PIV identity credentials such that client-application programs, compliant card applications, and compliant integrated circuits cards (ICC) can be used interchangeably by all information processing systems across Federal agencies. SP 800-73-3 defines the PIV data elements' identifiers, structure, and format. SP 800-73-3 also describes the client application programming interface and card command interface for use with the PIV Card.

This part, SP 800-73-3, Part 1 – *End-Point PIV Card Application Namespace, Data Model and Representation*, specifies the End-Point PIV Card Application Namespace, the PIV Data Model and its logical representation on the PIV Card, and is a companion document to FIPS 201.

1.4 Audience and Assumptions

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

1.5 Document Overview and Structure

All sections in this document are *normative* (i.e., mandatory for compliance) unless specified as *informative* (i.e., non-mandatory). Following is the structure of this document:

+ Section 1, *Introduction*, provides the purpose, scope, audience, and assumptions of the document and outlines its structure.
+ Section 2, *PIV Card Application Namespaces*, defines the three NIST managed namespaces used by the PIV Card Application.
+ Section 3, *End-Point PIV Data Model Elements*, describes the PIV Data Model elements in detail.
+ Section 4, *End-Point PIV Data Objects Representation*, describes the format and coding of the PIV data structures used by the PIV client-application programming interface and the PIV Card Application.
+ Section 5, *End-Point Data Types and Their Representation*, provides the details of the data types found on the PIV client-application programming interface and the PIV Card Application card command interface.
+ The appendices are informative and contain material that needs special formatting together with illustrative material to aid in understanding information in the body of the document.

2. PIV Card Application Namespaces

2.1 Namespaces of the PIV Card Application

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

+ Proprietary Identifier eXtension (PIX) of the NIST Registered Application Provider IDentifier (RID)
+ ASN.1 object identifiers (OIDs) in the personal verification subset of the OIDs managed by NIST
+ Basic Encoding Rules – Tag Length Value (BER-TLV) tags of the NIST PIV coexistent tag allocation scheme

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

All interindustry tags defined in ISO/IEC 7816, *Information Technology – Identification Cards – Integrated Circuit(s) Card with Contacts* [2], and used in the NIST coexistent tag allocation scheme without redefinition have the same meaning in the NIST PIV coexistent tag allocation scheme as they have in [2].

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

+ algorithm identifiers
+ key reference values
+ cryptographic mechanism identifiers

2.2 PIV Card Application AID

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

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

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

The PIV Card Application can be selected as the current application by providing the full AID as listed above or by providing the right-truncated version; that is, without the two-byte version, as follows:

'A0 00 00 03 08 00 00 10 00'

3. End-Point PIV Data Model Elements

This section contains the description of the data elements for personal identity verification, the PIV data model.

A PIV Card Application shall contain five mandatory interoperable data objects and may contain twenty-eight optional interoperable data objects. The five mandatory data objects for interoperable use are as follows:

1. Card Capability Container
2. Card Holder Unique Identifier
3. X.509 Certificate for PIV Authentication
4. Cardholder Fingerprints
5. Security Object

The twenty-eight optional data objects for interoperable use are as follows:

1. Cardholder Facial Image
2. Printed Information
3. X.509 Certificate for Digital Signature
4. X.509 Certificate for Key Management
5. X.509 Certificate for Card Authentication
6. Discovery Object
7. Key History Object
8. 20 retired X.509 Certificates for Key Management
9. Cardholder Iris Images

3.1 Mandatory Data Elements

The five mandatory data objects support FIPS 201 minimum mandatory compliance.

3.1.1 Card Capability Container

The Card Capability Container (CCC) is a mandatory data object whose purpose is to facilitate compatibility of GSC-IS applications with End-Point PIV Cards.

The CCC supports minimum capability for retrieval of the data model and optionally the application information as specified in Government Smart Card Interoperability Specification (GSC-IS) [3]. The data model of the PIV Card Application shall be identified by data model number 0x10. Deployed applications use 0x00 through 0x04. This enables the GSC-IS application domain to correctly identify a new data model namespace and structure as defined in this document.

For End-Point PIV Card Applications, the PIV data objects exist in a namespace tightly managed by NIST and a CCC discovery mechanism is not needed by End-Point applications. Therefore, all data elements of the CCC, except for the data model number, may optionally have a length value set to zero bytes (i.e., no value field will be supplied). The content of the CCC data elements, other than the data model number, are out of scope for this specification.

3.1.2 Card Holder Unique Identifier

The Card Holder Unique Identifier (CHUID) data object is defined in accordance with the Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems (TIG SCEPACS) [4]. For this specification, the CHUID is common between the contact and contactless chips. For dual chip implementations, the CHUID is copied in its entirety between the two chips.

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

+ The Buffer Length field is an optional TLV element. This element is the length in bytes of the entire CHUID, excluding the Buffer Length element itself, but including the CHUID's Asymmetric Signature element. The calculation of the asymmetric signature must exclude the Buffer Length element if it is present.

+ The Federal Agency Smart Credential Number (FASC-N) shall be in accordance with TIG SCEPACS [4]. A subset of the FASC-N, the FASC-N Identifier, shall be the unique identifier as described in [4, 6.6]: "The combination of an Agency Code, System Code, and Credential Number is a fully qualified number that is uniquely assigned to a single individual". The Agency Code is assigned to each Department or Agency by Special Publication 800-87 (SP 800-87), *Codes for Identification of Federal and Federally-Assisted Organizations* [5]. The subordinate System Code and Credential Number value assignment is subject to Department or Agency policy, provided that the FASC-N identifier (i.e., the concatenated Agency Code, System Code, and Credential Number) is unique for each card. The same FASC-N value shall be used in all the PIV data objects that include the FASC-N. To eliminate unnecessary use of the SSN[2], the FASC-N's Person Identifier (PI) field should not encode the SSN. TIG SCEPACS also specifies PACS interoperability requirements in Section 2.1, 10th paragraph of [4, 2.1]: "For full interoperability of a PACS it must at a minimum be able to distinguish fourteen digits (i.e., a combination of an Agency Code, System Code, and Credential Number) when matching FASC-N based credentials to enrolled card holders."

+ The Global Unique Identification number (GUID) field must be present, and shall include a UUID (see Section 3.3), an issuer assigned IPv6 address[3], or be coded as all zeros (0x00).

+ The DUNS and Organizational Code fields are optional.

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

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

3.1.3 X.509 Certificate for PIV Authentication

The X.509 Certificate for PIV Authentication and its associated private key, as defined in FIPS 201, is used to authenticate the card and the cardholder. The read access control rule for the X.509

[2] See the attachment to OMB M-07-16, Section 2: "Reduce the Use of Social Security Numbers".
[3] The use of IPv6 addresses in the GUID field is deprecated. It will be removed in a future revision of SP 800-73.

Certificate for PIV Authentication is “Always,” meaning the certificate can be read without access control restrictions. The Public Key Infrastructure (PKI) cryptographic function (see Table 3) is protected with a "PIN" access rule. In other words, private key operations using the PIV Authentication Key require the Personal Identification Number (PIN) to be submitted, but a successful PIN submission enables multiple private key operations without additional cardholder consent.

3.1.4 Cardholder Fingerprints

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

3.1.5 Security Object

The Security Object is in accordance with Appendix C of PKI for Machine Readable Travel Documents (MRTD) Offering ICC Read-Only Access Version 1.1 [6]. Tag 0xBA is used to map the ContainerIDs in the PIV data model to the 16 Data Groups specified in the MRTD. The mapping enables the Security Object to be fully compliant for future activities with identity documents.

The “DG-number-to-Container-ID” mapping object TLV in tag 0xBA encapsulates a series of three byte triples - one for each PIV data object included in the Security Object. The first byte is the Data Group (DG) number, and the second and third bytes are the most and least significant bytes (respectively) of the Container ID value. The DG number assignment is arbitrary; however, the same number assignment applies to the DataGroupNumber(s) in the DataGroupHash(es). This will ensure that the ContainerIDs in the mapping object refer to the correct hash values in the Security Object (0xBB).

The 0xBB Security Object is formatted according to the MRTD [6, Appendix C]. The LDS Security Object itself must be in ASN.1 DER format, formatted as specified in [6, Appendix C.2]. This structure is then inserted into the encapContentInfo field of the Cryptographic Message Syntax (CMS) object specified in [6, Appendix C.1].

The card issuer's digital signature key used to sign the CHUID shall also be used to sign the Security Object. The signature field of the Security Object, tag 0xBB, shall omit the issuer’s certificate, since it is included in the CHUID. At a minimum, unsigned data objects, such as the Printed Information data object, shall be included in the Security Object if present. For maximum protection against credential splicing attacks (credential substitution), it is recommended, however, that all PIV data objects, except the PIV X.509 certificates, be included in the Security Object.

3.2 Optional Data Elements

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

3.2.1 Cardholder Facial Image

The photo on the chip supports human verification only. It is not intended to support facial recognition systems for automated identity verification.

3.2.2 Printed Information

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

3.2.3 X.509 Certificate for Digital Signature

The X.509 Certificate for Digital Signature and its associated private key, as defined in FIPS 201, support the use of digital signatures for the purpose of document signing. The read access control rule for the X.509 Certificate is "Always", meaning the certificate can be read without access control restrictions. The PKI cryptographic function is protected with a "PIN Always" access rule. In other words, the PIN must be submitted every time immediately before a *Digital Signature Key* operation. This ensures cardholder participation every time the private key is used for digital signature generation.

3.2.4 X.509 Certificate for Key Management

The X.509 Certificate for Key Management and its associated private key, as defined in FIPS 201, support the use of encryption for the purpose of confidentiality. This key pair may be escrowed by the issuer for key recovery purposes. The read access control rule for the X.509 Certificate is "Always", meaning the certificate can be read without access control restrictions. The PKI cryptographic function is protected with a "PIN" access rule. In other words, once the PIN is submitted, subsequent *Key Management Key* operations can be performed without requiring the PIN again. This enables multiple private key operations without additional cardholder consent.

3.2.5 X.509 Certificate for Card Authentication

FIPS 201 specifies the optional Card Authentication Key (CAK) as an asymmetric or symmetric key that is used to support additional physical access applications. For an asymmetric CAK, the read access control rule of the corresponding X.509 Certificate for Card Authentication is "Always", meaning the certificate can be read without access control restrictions. Private (asymmetric) key operations or secret (symmetric) key operations are defined as "Always". In other words, the private or secret key can be used without access control restrictions. If the CAK is implemented, an asymmetric or symmetric CAK is generated by the PIV Card Issuer in accordance with FIPS 140-2 requirements for key generation. A CAK may be generated on-card or off-card. If a CAK is generated off-card, the result of each key generation will be injected into at most one PIV Card.

3.2.6 Discovery Object

The Discovery Object, if implemented, is the 0x7E interindustry ISO/IEC 7816-6 template that nests interindustry data objects. For the Discovery Object, the 0x7E template nests two BER-TLV structured interindustry data elements: 1) tag 0x4F contains the AID of the PIV Card Application and 2) tag 0x5F2F lists the PIN Usage Policy.

+ Tag 0x4F encodes the PIV Card Application AID as follows:

  {'4F 0B A0 00 00 03 08 00 00 10 00 01 00'}

+ Tag 0x5F2F encodes the PIN Usage Policy as follows:

First byte: 0x40 indicates that the PIV Card Application PIN alone satisfies the PIV Access Control Rules (ACRs) for command execution[4] and object access.

0x60 indicates that both the PIV Card Application PIN and Global PIN satisfy the PIV ACRs for command execution and PIV data object access.

Bits 5 through 1 of the first byte are RFU.

The second byte of the PIN Usage Policy encodes the cardholder's PIN preference for PIV Cards with both the PIV Card Application PIN and the Global PIN enabled:

Second byte: 0x10 indicates that the PIV Card Application PIN is the primary PIN used to satisfy the PIV ACRs for command execution and object access.

0x20 indicates that the Global PIN is the primary PIN used to satisfy the PIV ACRs for command execution and object access.

Note: If the first byte is set to 0x40, then the second byte is RFU and shall be set to 0x00.

PIV Card Applications that satisfy the PIV ACRs for PIV data object access and command execution[5] with both the PIV Card Application PIN and Global PIN shall implement the Discovery Object with the PIN Usage Policy set to 0x60 zz where zz is set to either 0x10 or 0x20.

The encoding of the 0x7E Discovery Object is as follows:

{'7E 12' {{'4F 0B A0 00 00 03 08 00 00 10 00 01 00'} {'5F 2F 02 xx yy'}}}, where xx and yy encode the first and second byte of the PIN Usage Policy as described in this section.

The Security Object enforces integrity of the Discovery Object according to the issuer.

3.2.7 Key History Object

Up to twenty retired Key Management private keys may be stored in the PIV Card Application. The Key History object provides information about the retired Key Management private keys that are present within the PIV Card Application. Retired Key Management private keys are private keys that correspond to X.509 certificates for Key Management that have expired, have been revoked, or have otherwise been superseded. The Key History object shall be present in the PIV Card Application if the PIV Card Application contains any retired Key Management private keys, but may be present even if no such keys are present in the PIV Card Application. For each retired Key Management private key in the PIV Card Application, the corresponding certificate may either be present within the PIV Card Application or may only be available from an on-line repository.

The Key History object includes two mandatory fields, *keysWithOnCardCerts* and *keysWithOffCardCerts*, and one optional field, *offCardCertURL*. The *keysWithOnCardCerts* field indicates the number of retired private keys within the PIV Card Application for which the corresponding certificates are also stored within the PIV Card Application. The *keysWithOffCardCerts* field indicates the number of retired private keys within the PIV Card

[4] Command execution pertains to the VERIFY APDU and optionally to the CHANGE REFERENCE DATA APDU.
[5] Command execution pertains to the VERIFY APDU and optionally to the CHANGE REFERENCE DATA APDU.

Application for which the corresponding certificates are not stored within the PIV Card Application. The numeric values in both *keysWithOnCardCerts* and *keysWithOffCardCerts* are represented as unsigned binary integers. The *offCardCertURL* field contains a URL that points to a file containing the certificates corresponding to all of the retired private keys within the PIV Card Application, including those for which the corresponding certificate is also stored within the PIV Card Application. The *offCardCertURL* field shall be present if the *keysWithOffCardCerts* value is greater than zero and shall be absent if the values of both *keysWithOnCardCerts* and *keysWithOffCardCerts* are zero. The *offCardCertURL* field may be present if the *keysWithOffCardCerts* value is zero but the *keysWithOnCardCerts* value is greater than zero.

The file that is pointed to by the *offCardCertURL* field shall contain the DER encoding of the following data structure:

```
OffCardKeyHistoryFile ::= SEQUENCE SIZE (1..20) OF SEQUENCE {
        keyReference        OCTET STRING (SIZE(1))
        cert                Certificate
}
```

where **keyReference** is the key reference for the private key on the card and **cert** is the corresponding X.509 certificate.[6] The *offCardCertURL* field shall have the following format:

"http://" <DNS name> "/" <ASCII-HEX encoded SHA-256 hash [14] of **OffCardKeyHistoryFile**>

The private keys for which the corresponding certificates are stored within the PIV Card Application shall be assigned to the lowest numbered key references reserved for retired Key Management private keys. For example if *keysWithOnCardCerts* is 5, then the corresponding private keys shall be assigned to key references '82', '83', '84', '85', and '86'.

The private keys for which the corresponding certificates are not stored within the PIV Card Application shall be assigned to the highest numbered key references reserved for retired Key Management private keys. For example, if *keysWithOffCardCerts* is 3, then the corresponding private keys shall be assigned to key references '93', '94', and '95'.

Private keys do not have to be stored within the PIV Card Application in the order of their age. However, if the certificates corresponding to only some of the retired Key Management private keys are available within the PIV Card Application then the certificates that are stored in the PIV Card Application shall be the ones that were most recently issued.

The Key History object is only available over the contact interface. The read access control rule for the Key History object is “Always”, meaning that it can be read without access control restrictions.

The Security Object enforces integrity of the Key History object according to the issuer.

3.2.8 Retired X.509 Certificates for Key Management

These objects hold the X.509 certificates for Key Management corresponding to retired Key Management Keys, as described in Section 3.2.7. Retired Key Management private keys and their corresponding certificates are only available over the contact interface. The read access control rule for these certificates is “Always”, meaning the certificates can be read without access control restrictions. The PKI cryptographic function for all of the retired Key Management Keys is protected

[6] The ASN.1 for **Certificate** may be imported from the ASN.1 module **PKIX1Explicit88** in Appendix A.1 of RFC 5280 [13].

with a "PIN" access rule. In other words, once the PIN is submitted and verified, subsequent *Key Management Key* operations can be performed with any of the retired Key Management Keys without requiring the PIN again. This enables multiple private key operations without additional cardholder consent.

3.2.9 Cardholder Iris Images

The iris data object specifies compact images of the cardholder's irises. The images are suitable for use in iris recognition systems for automated identity verification.

3.3 Inclusion of Universally Unique IDentifiers (UUIDs)

As defined in [10], the presence of a Universally Unique IDentifier (UUID) conformant to the specification [11] is required in each identification card issued by Non-Federal Issuers, referred to as "PIV Interoperable" (PIV-I) or "PIV Compatible" (PIV-C) cards. The intent of [10] is to enable issuers to issue cards that are technically interoperable with Federal PIV Card readers and applications, and that may be trusted for particular purposes through a decision of the relying Federal Department or Agency. Because the goal is interoperability of PIV-I and PIV-C cards with the Federal PIV System, the technical requirements for the inclusion of the UUID are specified in this document. To include a UUID identifier on a PIV-I, PIV-C, or PIV Card, a credential issuer shall meet the following specifications for all relevant data objects present on an issued identification card.

1. If the card is a PIV-I or PIV-C card, the FASC-N in the CHUID shall have Agency Code equal to 9999, System Code equal to 9999, and Credential Number equal to 999999, indicating that a UUID is the primary credential identifier. In this case, the FASC-N shall be omitted from certificates and CMS-signed data objects. If the card is a PIV Card, the FASC-N in the CHUID shall be populated as described in Section 3.1.2, and the FASC-N shall be included in authentication certificates and CMS-signed data objects as required by FIPS 201.

2. The value of the GUID data element of the CHUID data object shall be a 16-byte binary representation of a valid UUID [11]. The UUID should be version 1, 4, or 5, as specified in [11], Section 4.1.3.

3. The same 16-byte binary representation of the UUID value shall be present as the value of an entryUUID attribute, as defined in [12], in any CMS-signed data object that is required to contain a pivFASC-N attribute on a PIV Card, i.e., in the fingerprint template and facial image data objects, if present.

4. The string representation of the same UUID value shall be present in the PIV Authentication Certificate and the Card Authentication Certificate, if present, in the subjectAltName extension encoded as a URI, as specified by [11], Section 3.

The option specified in this section supports the use of UUIDs by Non-Federal Issuers. It also allows, but does not require, the use of UUIDs as optional data elements on PIV Cards. PIV Cards must meet all requirements in FIPS 201 whether or not the UUID identifier option is used; in particular, the FASC-N identifier must be present in all PIV data objects as specified by FIPS 201 and its normative references. PIV Cards that include UUIDs must include the UUIDs in all data objects described in (2) through (4).

3.4 Data Object Containers and associated Access Rules and Interface Modes

Table 1 defines a high level view of the data model. Each on-card storage container is labeled either as Mandatory (M) or Optional (O). This data model is designed to enable and support dual interface cards. Note that access conditions based on the interface mode (contact vs. contactless) take precedence over all Access Rules defined in Table 1, Column 3.

Table 1. Data Model Containers

Container Name	Container ID	Access Rule for Read	Contact / Contactless[7]	M/O
Card Capability Container	0xDB00	Always	Contact	M
Card Holder Unique Identifier	0x3000	Always	Contact and Contactless	M
X.509 Certificate for PIV Authentication	0x0101	Always	Contact	M
Cardholder Fingerprints	0x6010	PIN	Contact	M
Security Object	0x9000	Always	Contact	M
Cardholder Facial Image	0x6030	PIN	Contact	O
Printed Information	0x3001	PIN	Contact	O
X.509 Certificate for Digital Signature	0x0100	Always	Contact	O
X.509 Certificate for Key Management	0x0102	Always	Contact	O
X.509 Certificate for Card Authentication	0x0500	Always	Contact and Contactless	O
Discovery Object	0x6050	Always	Contact and Contactless	O
Key History Object	0x6060	Always	Contact	O
Retired X.509 Certificate for Key Management 1	0x1001	Always	Contact	O
Retired X.509 Certificate for Key Management 2	0x1002	Always	Contact	O
Retired X.509 Certificate for Key Management 3	0x1003	Always	Contact	O
Retired X.509 Certificate for Key Management 4	0x1004	Always	Contact	O
Retired X.509 Certificate for Key Management 5	0x1005	Always	Contact	O
Retired X.509 Certificate for Key Management 6	0x1006	Always	Contact	O
Retired X.509 Certificate for Key Management 7	0x1007	Always	Contact	O
Retired X.509 Certificate for Key Management 8	0x1008	Always	Contact	O
Retired X.509 Certificate for Key Management 9	0x1009	Always	Contact	O
Retired X.509 Certificate for Key Management 10	0x100A	Always	Contact	O
Retired X.509 Certificate for Key Management 11	0x100B	Always	Contact	O
Retired X.509 Certificate for Key Management 12	0x100C	Always	Contact	O
Retired X.509 Certificate for Key Management 13	0x100D	Always	Contact	O
Retired X.509 Certificate for Key Management 14	0x100E	Always	Contact	O
Retired X.509 Certificate for Key Management 15	0x100F	Always	Contact	O
Retired X.509 Certificate for Key Management 16	0x1010	Always	Contact	O
Retired X.509 Certificate for Key Management 17	0x1011	Always	Contact	O
Retired X.509 Certificate for Key Management 18	0x1012	Always	Contact	O
Retired X.509 Certificate for Key Management 19	0x1013	Always	Contact	O

[7] Contact interface mode means the container is accessible through contact interface only. Contact and contactless interface mode means the container can be accessed from either interface.

Retired X.509 Certificate for Key Management 20	0x1014	Always	Contact	O
Cardholder Iris Image	0x1015	PIN	Contact	O

Appendix A provides a detailed spreadsheet for the data model. ContainerIDs and Tags within the containers for each data object are defined by this data model in accordance with SP 800-73-3 naming conventions.

4. End-Point PIV Data Objects Representation

4.1 Data Objects Definition

A *data object* is an item of information seen on the card command interface for which is 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* (OID), as defined in ISO/IEC 8824-2:2002. [7]

A data object whose data content is encoded as a BER-TLV data structure as in ISO/IEC 8825—1:2002 [8] is called a *BER-TLV data object.*

4.1.1 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 the 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.

The PIV End-Point Data objects are BER-TLV objects encoded as per [8], except that Tag values of the PIV data object's inner tag assignments do not conform to BER-TLV requirements.[8] This is due to the need to accommodate legacy tags inherited from the GSC-IS.

4.2 OIDs and Tags of PIV Card Application Data Objects

Table 2 lists the ASN.1 object identifiers and BER-TLV tags of the thirty-three PIV Card Application data objects for interoperable use. For the purpose of constructing PIV Card Application data object names in the CardApplicationURL in the CCC of the PIV Card Application, the NIST RID ('A0 00 00 03 08') shall be used and the card application type shall be set to '00'.

4.3 Object Identifiers

Each of the data objects in the PIV Card Application has been provided with a three-byte BER-TLV tag and an ASN.1 OID from the NIST personal identity verification arc. These object identifier assignments are given in Table 2.

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

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

[8] The exception does not apply to the Discovery Object, nor the Application Property Template (APT), since these objects use interindustry tags from ISO/IEC 7816-6.

Table 1 lists the ACRs of the thirty-three PIV Card Application data objects for interoperable use. See Table 3 in Section 5.1 and Table 6-3 in Special Publication 800-78 [9], for the key references and permitted algorithms associated with these authenticatable entities.

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

Data Object for Interoperable Use	ASN.1 OID	BER-TLV Tag	M/O
Card Capability Container	2.16.840.1.101.3.7.1.219.0	'5FC107'	M
Card Holder Unique Identifier	2.16.840.1.101.3.7.2.48.0	'5FC102'	M
X.509 Certificate for PIV Authentication	2.16.840.1.101.3.7.2.1.1	'5FC105'	M
Cardholder Fingerprints	2.16.840.1.101.3.7.2.96.16	'5FC103'	M
Security Object	2.16.840.1.101.3.7.2.144.0	'5FC106'	M
Cardholder Facial Image	2.16.840.1.101.3.7.2.96.48	'5FC108'	O
Printed Information	2.16.840.1.101.3.7.2.48.1	'5FC109'	O
X.509 Certificate for Digital Signature	2.16.840.1.101.3.7.2.1.0	'5FC10A'	O
X.509 Certificate for Key Management	2.16.840.1.101.3.7.2.1.2	'5FC10B'	O
X.509 Certificate for Card Authentication	2.16.840.1.101.3.7.2.5.0	'5FC101'	O
Discovery Object	2.16.840.1.101.3.7.2.96.80	'7E'	O
Key History Object	2.16.840.1.101.3.7.2.96.96	'5FC10C'	O
Retired X.509 Certificate for Key Management 1	2.16.840.1.101.3.7.2.16.1	'5FC10D'	O
Retired X.509 Certificate for Key Management 2	2.16.840.1.101.3.7.2.16.2	'5FC10E'	O
Retired X.509 Certificate for Key Management 3	2.16.840.1.101.3.7.2.16.3	'5FC10F'	O
Retired X.509 Certificate for Key Management 4	2.16.840.1.101.3.7.2.16.4	'5FC110'	O
Retired X.509 Certificate for Key Management 5	2.16.840.1.101.3.7.2.16.5	'5FC111'	O
Retired X.509 Certificate for Key Management 6	2.16.840.1.101.3.7.2.16.6	'5FC112'	O
Retired X.509 Certificate for Key Management 7	2.16.840.1.101.3.7.2.16.7	'5FC113'	O
Retired X.509 Certificate for Key Management 8	2.16.840.1.101.3.7.2.16.8	'5FC114'	O
Retired X.509 Certificate for Key Management 9	2.16.840.1.101.3.7.2.16.9	'5FC115'	O
Retired X.509 Certificate for Key Management 10	2.16.840.1.101.3.7.2.16.10	'5FC116'	O
Retired X.509 Certificate for Key Management 11	2.16.840.1.101.3.7.2.16.11	'5FC117'	O
Retired X.509 Certificate for Key Management 12	2.16.840.1.101.3.7.2.16.12	'5FC118'	O
Retired X.509 Certificate for Key Management 13	2.16.840.1.101.3.7.2.16.13	'5FC119'	O
Retired X.509 Certificate for Key Management 14	2.16.840.1.101.3.7.2.16.14	'5FC11A'	O
Retired X.509 Certificate for Key Management 15	2.16.840.1.101.3.7.2.16.15	'5FC11B'	O
Retired X.509 Certificate for Key Management 16	2.16.840.1.101.3.7.2.16.16	'5FC11C'	O
Retired X.509 Certificate for Key Management 17	2.16.840.1.101.3.7.2.16.17	'5FC11D'	O
Retired X.509 Certificate for Key Management 18	2.16.840.1.101.3.7.2.16.18	'5FC11E'	O
Retired X.509 Certificate for Key Management 19	2.16.840.1.101.3.7.2.16.19	'5FC11F'	O
Retired X.509 Certificate for Key Management 20	2.16.840.1.101.3.7.2.16.20	'5FC120'	O
Cardholder Iris Images	2.16.840.1.101.3.7.2.16.21	'5FC121'	O

5. End-Point Data Types and Their Representation

This section provides a description of the data types used in the PIV Client Application Programming Interface (SP 800-73-3, Part 3) and PIV Card Command Interface (SP 800-73-3, Part 2). Unless otherwise indicated, the representation shall be the same on both interfaces.

The data types are defined in Part 1, rather than in Parts 2 and 3 in order to achieve smart card platform independence from Part 1. Thus, non-government smart card programs can readily adopt the interface specifications in Parts 2 and 3 while customizing Part 1 to their own data model, data types, and namespaces.

5.1 Key References

A key reference is a one-byte reference data identifier that specifies a cryptographic key or PIN according to its PIV Key Type. Table 3 and SP 800-78, Table 6-1, define the key reference values that shall be used on the PIV interfaces. The key reference values are used, for example, in a cryptographic protocol such as an authentication or a signing protocol. Key references are only assigned to private and secret (symmetric) keys and PINs. All other PIV Card Application key reference values are reserved for future use.

Table 3. PIV Card Application Authentication and Key References

Key Reference Value	PIV Key Type	Authenticatable Entity / Administrator	Security Condition for Use	Retry Reset Value	Number of Unblocks
'00'	Global PIN	Cardholder	Always	Platform Specific	Platform Specific
'80'	PIV Card Application PIN	Cardholder	Always	Issuer Specific	Issuer Specific
'81'	PIN Unblocking Key	PIV Card Application Administrator	Always	Issuer Specific	Issuer Specific
See Table 6-1 in SP 800-78	*PIV Authentication Key*	PIV Card Application Administrator	PIN	N/A	N/A
See Table 6-1 in SP 800-78	*Card Management Key*[9]	PIV Card Application Administrator	Always	N/A	N/A
See Table 6-1 in SP 800-78	*Digital Signature Key*	PIV Card Application Administrator	PIN Always	N/A	N/A
See Table 6-1 in SP 800-78	*Key Management Key*	PIV Card Application Administrator	PIN	N/A	N/A
See Table 6-1 in SP 800-78	*Card Authentication Key*	PIV Card Application Administrator	Always	N/A	N/A

[9] Note: The Card Management key is the PIV Card Application Administration Key used for managing the PIV Card Application.

Key Reference Value	PIV Key Type	Authenticatable Entity / Administrator	Security Condition for Use	Retry Reset Value	Number of Unblocks
'82', '83', '84', '85', '86', '87', '88', '89', '8A', '8B', '8C', '8D', '8E', '8F', '90', '91', '92', '93', '94', '95'	*Retired Key Management Key*	PIV Card Application Administrator	PIN	N/A	N/A

When represented as a byte, the key reference occupies bits b8 and b5-b1, while b7 and b6 shall be set to 0. If b8 is 0 then the key reference names global reference data. If b8 is 1, then the key reference names application-specific reference data.

The access control rules for PIV data object access shall reference the PIV Card Application PIN and may optionally reference the cardholder Global PIN. If the Global PIN is used by the PIV Card Application then the Global PIN format shall follow the PIV Card Application PIN format defined in Section 2.4.3 of Part 2.

PIV Card Applications with the Discovery Object, and the first byte of the PIN Usage Policy value set to 0x60 as per Section 3.2.6, shall reference the PIV Card Application PIN as well as the cardholder Global PIN in the access control rules for PIV data object access. Additionally, the PIV Card Application card commands can change the status of the Global PIN, and may change its reference data while the PIV Card Application is the currently selected application.

Note: The rest of the document uses “PIN” to mean either the PIV Application PIN or the Global PIN.

5.2 PIV Algorithm Identifier

A PIV algorithm identifier is a one-byte identifier of a cryptographic algorithm. The identifier specifies a cryptographic algorithm and key size. For symmetric cryptographic operations, the algorithm identifier also specifies a mode of operation (ECB). SP 800-78, Table 6-2 lists the PIV algorithm identifiers for the cryptographic algorithms that may be recognized on the PIV interfaces.

5.3 Cryptographic Mechanism Identifiers

Cryptographic Mechanism Identifiers are defined in Table 4. These identifiers serve as data field inputs to the SP 800-73-3 Part 2 GENERATE ASYMMETRIC KEY PAIR card command and the SP 800-73-3 Part 3 pivGenerateKeyPair() client API function call, which initiates the generation and storage of the asymmetric key pair.

Table 4. Cryptographic Mechanism Identifiers

Cryptographic Mechanism Identifier	Description	Parameter
'00'-'05'	RFU	
See Table 6-2 in SP 800-78	RSA 1024	Optional public exponent encoded big-endian
See Table 6-2 in SP 800-78	RSA 2048	Optional public exponent encoded

		big-endian
'08'-'10'	RFU	
See Table 6-2 in SP 800-78	ECC: Curve P-256	None
'12'-'13'	RFU	
See Table 6-2 in SP 800-78	ECC: Curve P-384	None

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

5.4 Status Words

A Status Word (SW) is a 2-byte value returned by 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 the card command interface and their interpretation are given in Table 5. The descriptions of individual card commands provide additional information for interpreting returned status words.

Table 5. Status Words

SW1	SW2	Meaning
'61'	'xx'	Successful execution where SW2 encodes the number of response data bytes still available
'63'	'CX'	Verification failed, X indicates the number of further allowed retries or resets
'69'	'82'	Security condition not satisfied
'69'	'83'	Authentication method blocked
'6A'	'80'	Incorrect parameter in command data field
'6A'	'81'	Function not supported
'6A'	'82'	Data object or application not found
'6A'	'84'	Not enough memory
'6A'	'86'	Incorrect parameter in P1 or P2
'6A'	'88'	Referenced data or reference data not found
'90'	'00'	Successful execution

Appendix A—PIV Data Model

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

The SP 800-73-3 End-Point specification does not provide mechanisms to read partial contents of a PIV data object. Individual access to the TLV elements within a container is not supported. For each container, End-Point compliant cards shall return all TLV elements of the container in the order listed in this Appendix.

Both single-chip/dual-interface and dual-chip implementations are be feasible. In the single-chip/dual-interface configuration, the PIV Card Application shall be provided the information regarding which interface is in use. In the dual-chip configuration, a separate PIV Card Application shall be loaded on each chip.

Table 6. PIV Data Containers

Container Description	Container ID	BER-TLV Tag	Container Minimum Capacity (Bytes)*	Access Rule for Read	Contact / Contactless	M/O
Card Capability Container	0xDB00	'5FC107'	297	Always	Contact	M
Card Holder Unique Identifier	0x3000	'5FC102'	2898	Always	Contact and Contactless	M
X.509 Certificate for PIV Authentication (Key Reference '9A')	0x0101	'5FC105'	2005	Always	Contact	M
Cardholder Fingerprints	0x6010	'5FC103'	4006	PIN	Contact	M
Security Object	0x9000	'5FC106'	1055	Always	Contact	M
Cardholder Facial Image	0x6030	'5FC108'	12710	PIN	Contact	O
Printed Information	0x3001	'5FC109'	142	PIN	Contact	O
X.509 Certificate for Digital Signature (Key Reference '9C')	0x0100	'5FC10A'	2005	Always	Contact	O
X.509 Certificate for Key Management (Key Reference '9D')	0x0102	'5FC10B'	2005	Always	Contact	O
X.509 Certificate for Card Authentication (Key Reference '9E')	0x0500	'5FC101'	2005	Always	Contact and Contactless	O
Discovery Object	0x6050	'7E'	20	Always	Contact and Contactless	O
Key History Object	0x6060	'5FC10C'	128	Always	Contact	O
Retired X.509 Certificate for Key Management 1 (Key reference '82')	0x1001	'5FC10D'	2005	Always	Contact	O

* The values in this column denote the guaranteed minimum capacities, in bytes, of the on-card storage containers. Cards with larger containers may be produced and determined conformant.

Retired X.509 Certificate for Key Management 2 (Key reference '83')	0x1002	'5FC10E'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 3 (Key reference '84')	0x1003	'5FC10F'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 4 (Key reference '85')	0x1004	'5FC110'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 5 (Key reference '86')	0x1005	'5FC111'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 6 (Key reference '87')	0x1006	'5FC112'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 7(Key reference '88')	0x1007	'5FC113'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 8(Key reference '89')	0x1008	'5FC114'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 9 (Key reference '8A')	0x1009	'5FC115'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 10 (Key reference '8B')	0x100A	'5FC116'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 11 (Key reference '8C')	0x100B	'5FC117'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 12 (Key reference '8D')	0x100C	'5FC118'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 13 (Key reference '8E')	0x100D	'5FC119'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 14 (Key reference '8F')	0x100E	'5FC11A'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 15 (Key reference '90')	0x100F	'5FC11B'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 16 (Key reference '91')	0x1010	'5FC11C'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 17 (Key reference '92')	0x1011	'5FC11D'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 18 (Key reference '93')	0x1012	'5FC11E'	2005	Always	Contact	O
Retired X.509 Certificate for Key Management 19 (Key reference '94')	0x1013	'5FC11F'	2005	Always	Contact	O

Retired X.509 Certificate for Key Management 20 (Key reference '95')	0x1014	'5FC120'	2005	Always	Contact	O
Cardholder Iris Images	0x1015	'5FC121'	7106	PIN	Contact	O

Note that all data elements of the following data objects are mandatory unless specified as optional.

Table 7. Card Capability Container

Card Capability Container	0xDB00		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Card Identifier	0xF0	Fixed	21
Capability Container version number	0xF1	Fixed	1
Capability Grammar version number	0xF2	Fixed	1
Applications CardURL	0xF3	Variable	128
PKCS#15	0xF4	Fixed	1
Registered Data Model number	0xF5	Fixed	1
Access Control Rule Table	0xF6	Fixed	17
Card APDUs	0xF7	Fixed	0
Redirection Tag	0xFA	Fixed	0
Capability Tuples (CTs)	0xFB	Fixed	0
Status Tuples (STs)	0xFC	Fixed	0
Next CCC	0xFD	Fixed	0
Extended Application CardURL (optional)	0xE3	Fixed	48
Security Object Buffer (optional)	0xB4	Fixed	48
Error Detection Code	0xFE	LRC	0

Table 8. Card Holder Unique Identifier

Card Holder Unique Identifier	0x3000		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Buffer Length (Optional)	0xEE	Fixed	2
FASC-N	0x30	Fixed Text	25
Organization Identifier (Optional)	0x32	Fixed	4
DUNS (Optional)	0x33	Fixed	9
GUID	0x34	Fixed Numeric	16
Expiration Date	0x35	Date (YYYYMMDD)	8
Issuer Asymmetric Signature	0x3E	Variable	2816**
Error Detection Code	0xFE	LRC	0

The Error Detection Code is the same element as the Longitudinal Redundancy Code (LRC) in TIG SCEPACS. Because TIG SCEPACS makes the LRC mandatory, it is present in the CHUID. However, this document makes no use of the Error Detection Code, and therefore the length of the TLV value is set to 0 bytes (i.e., no value will be supplied).

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.
** Recommended length: The signer certificate may cause the "Max. Bytes" value in the Issuer Asymmetric Signature field to be exceeded.

Note: The Authentication Key Map data element has been removed from Table 8 as it has been previously deprecated.

Table 9. X.509 Certificate for PIV Authentication

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

Table 10. Cardholder Fingerprints

Cardholder Fingerprints	0x6010		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Fingerprint I & II	0xBC	Variable	4000***
Error Detection Code	0xFE	LRC	0

Table 11. Security Object

Security Object	0x9000		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Mapping of DG to ContainerID	0xBA	Variable	100
Security Object	0xBB	Variable	900
Error Detection Code	0xFE	LRC	0

Table 12. Cardholder Facial Image

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

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.
** Recommended length. Certificate size can exceed indicated length value.
*** Recommended length. The certificate that signed the Fingerprint I and II data element in the Cardholder Fingerprint data object can either be stored in the CHUID or in the Fingerprint I and II data element itself. If the latter, the "Max. Bytes" value quoted is a recommendation and the signer certificate in CBEFF_SIGNATURE_BLOCK can exceed the "Max. bytes".
**** Recommended length. The certificate that signed the Facial Image data element (tag 0xBC) can be stored in the CHUID or in the Facial Image data object itself. If the latter, the "Max. Bytes" value quoted is a recommendation and the signer certificate in CBEFF_SIGNATURE_BLOCK can exceed the "Max. bytes".

Table 13. Printed Information

Printed Information	0x3001		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Name	0x01	Fixed Text	32
Employee Affiliation	0x02	Fixed Text	20
Expiration date	0x04	Date (YYYYMMMDD)	9
Agency Card Serial Number	0x05	Fixed Text	10
Issuer Identification	0x06	Fixed Text	15
Organization Affiliation (Line 1) (Optional)	0x07	Fixed Text	20
Organization Affiliation (Line 2) (Optional)	0x08	Fixed Text	20
Error Detection Code	0xFE	LRC	0

Note: The previously deprecated Employee Affiliation Line 2 data element (tag 0x03) has been eliminated, as it did not have a corresponding text field on the face of the card. In order to successfully match the printed information for verification on Zone 8 (Employee Affiliation) and Zone 10 (Organization Affiliation) on the face of the card with the printed information stored electronically on the card, agencies should use tags 0x02, 0x07 and 0x08.

Table 14. X.509 Certificate for Digital Signature

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

Table 15. X.509 Certificate for Key Management

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

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.
** Recommended length. Certificate size can exceed indicated length value.

Table 16. X.509 Certificate for Card Authentication

X.509 Certificate for Card Authentication	0x0500		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 17. Discovery Object

Discovery Object (Tag '7E')	0x6050		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
PIV Card Application AID	0x4F	Fixed	12
PIN Usage Policy	0x5F2F	Fixed	3

Table 18. Key History Object

Key History Object	0x6060		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
keysWithOnCardCerts	0xC1	Fixed	1
keysWithOffCardCerts	0xC2	Fixed	1[11]
offCardCertURL (Conditional)***	0xF3	Variable	118
Error Detection Code	0xFE	LRC	0

Table 19. Retired X.509 Certificate for Key Management 1

Retired X.509 Certificate for Key Management 1	0x1001		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

*** The offCardCertURL data element shall be present if keysWithOffCardCerts is greater than zero and shall be absent if both keysWithOnCardCerts and keysWithOffCardCerts are zero. The offCardCertURL may be present if keyWithOffCardCerts is zero but keysWithOnCardCerts is greater than zero.

[11] The numeric values indicated in keysWithOnCardCerts and keysWithOffCardCerts are represented as unsigned binary integers.

Table 20. Retired X.509 Certificate for Key Management 2

Retired X.509 Certificate for Key Management 2	0x1002		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 21. Retired X.509 Certificate for Key Management 3

Retired X.509 Certificate for Key Management 3	0x1003		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 22. Retired X.509 Certificate for Key Management 4

Retired X.509 Certificate for Key Management 4	0x1004		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 23. Retired X.509 Certificate for Key Management 5

Retired X.509 Certificate for Key Management 5	0x1005		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 24. Retired X.509 Certificate for Key Management 6

Retired X.509 Certificate for Key Management 6	0x1006		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

Retired X.509 Certificate for Key Management 6		0x1006	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 25. Retired X.509 Certificate for Key Management 7

Retired X.509 Certificate for Key Management 7		0x1007	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 26. Retired X.509 Certificate for Key Management 8

Retired X.509 Certificate for Key Management 8		0x1008	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 27. Retired X.509 Certificate for Key Management 9

Retired X.509 Certificate for Key Management 9		0x1009	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 28. Retired X.509 Certificate for Key Management 10

Retired X.509 Certificate for Key Management 10		0x100A	
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

Table 29. Retired X.509 Certificate for Key Management 11

Retired X.509 Certificate for Key Management 11	0x100B		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 30. Retired X.509 Certificate for Key Management 12

Retired X.509 Certificate for Key Management 12	0x100C		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 31. Retired X.509 Certificate for Key Management 13

Retired X.509 Certificate for Key Management 13	0x100D		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 32. Retired X.509 Certificate for Key Management 14

Retired X.509 Certificate for Key Management 14	0x100E		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 33. Retired X.509 Certificate for Key Management 15

Retired X.509 Certificate for Key Management 15	0x100F		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

Retired X.509 Certificate for Key Management 15	0x100F		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 34. Retired X.509 Certificate for Key Management 16

Retired X.509 Certificate for Key Management 16	0x1010		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 35. Retired X.509 Certificate for Key Management 17

Retired X.509 Certificate for Key Management 17	0x1011		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 36. Retired X.509 Certificate for Key Management 18

Retired X.509 Certificate for Key Management 18	0x1012		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

Table 37. Retired X.509 Certificate for Key Management 19

Retired X.509 Certificate for Key Management 19	0x1013		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

Table 38. Retired X.509 Certificate for Key Management 20

Retired X.509 Certificate for Key Management 20	0x1014		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Certificate	0x70	Variable	1856**
CertInfo	0x71	Fixed	1
MSCUID (Optional)	0x72	Variable	38
Error Detection Code	0xFE	LRC	0

The CertInfo byte in the certificate data objects identified above shall be encoded as follows:

b8	b7	b6	b5	b4	b3	b2	b1
RFU8	RFU7	RFU6	RFU5	RFU4	IsX509	CompressionTypeLsb	CompressionTypeMsb

CompressionTypeMsb shall be 0 if the certificate is encoded in uncompressed form and 1 if the certificate is encoded using GZIP compression.[13] CompressionTypeLsb and IsX509 shall be set to 0 for PIV Applications. Thus, for a certificate encoded in uncompressed form CertInfo shall be '0x00', and for a certificate encoded using GZIP compression CertInfo shall be '0x01'.

Table 39. Cardholder Iris Images

Cardholder Iris Images	0x1015		
Data Element (TLV)	**Tag**	**Type**	**Max. Bytes***
Images for Iris	0xBC	Variable	7100****
Error Detection Code	0xFE	LRC	0

* The values in the "Max. Bytes" columns denote the lengths of the value (V) fields of BER-TLV elements.

** Recommended length. Certificate size can exceed indicated length value.

[13] GZIP formats are specified in RFC 1951 and RFC 1952.

**** Recommended length. The certificate that signed the Images for Iris data element (tag 0xBC) can be stored in the CHUID or in the Images for Iris data object itself. If the latter, the "Max. Bytes" value quoted is a recommendation and the signer certificate in CBEFF_SIGNATURE_BLOCK can exceed the "Max. bytes".

Appendix B—PIV Authentication Mechanisms

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

Card Validation (CardV) — This is the process of verifying that a PIV Card is authentic (i.e., not a counterfeit card). Card validation mechanisms include:

+ visual inspection of the tamper-proofing and tamper-resistant features of the PIV Card as per Section 4.1.2 of FIPS 201;
+ use of cryptographic challenge-response schemes with symmetric keys; and
+ use of asymmetric authentication schemes to validate private keys embedded within the PIV Card.

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

+ visual inspection of PIV Card visual elements (such as the photo, the printed name, and rank, if present);
+ verification of certificates on the PIV Card;
+ verification of signatures on the PIV biometrics and the CHUID;
+ checking the expiration date; and
+ checking the revocation status of the credentials on the PIV Card.

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

+ presentation of a PIV Card by the cardholder;
+ matching the visual characteristics of the cardholder with the photo on the PIV Card;
+ matching the PIN provided with the PIN on the PIV Card; and

+ matching the live fingerprint samples provided by the cardholder with the biometric information embedded within the PIV Card.

B.1 Authentication Mechanism Diagrams

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

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

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

B.1.1 Authentication using PIV Visual Credentials

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

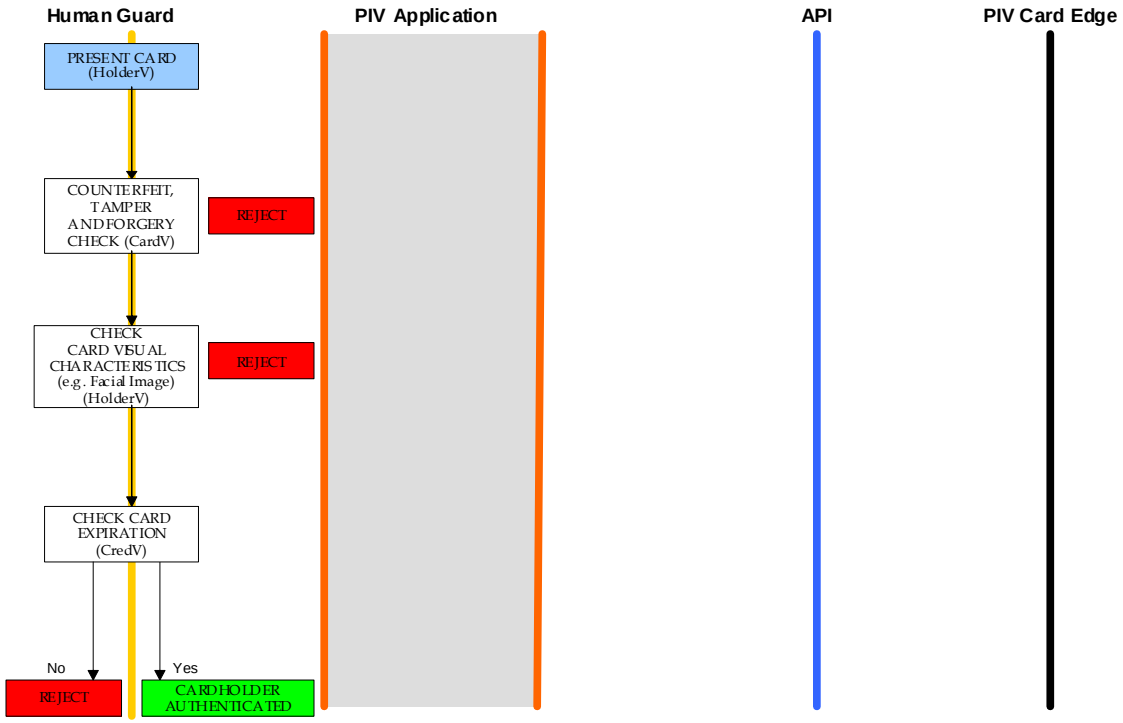

Figure B-1. Authentication using PIV Visual Credentials

B.1.2 Authentication using PIV *CHUID*

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

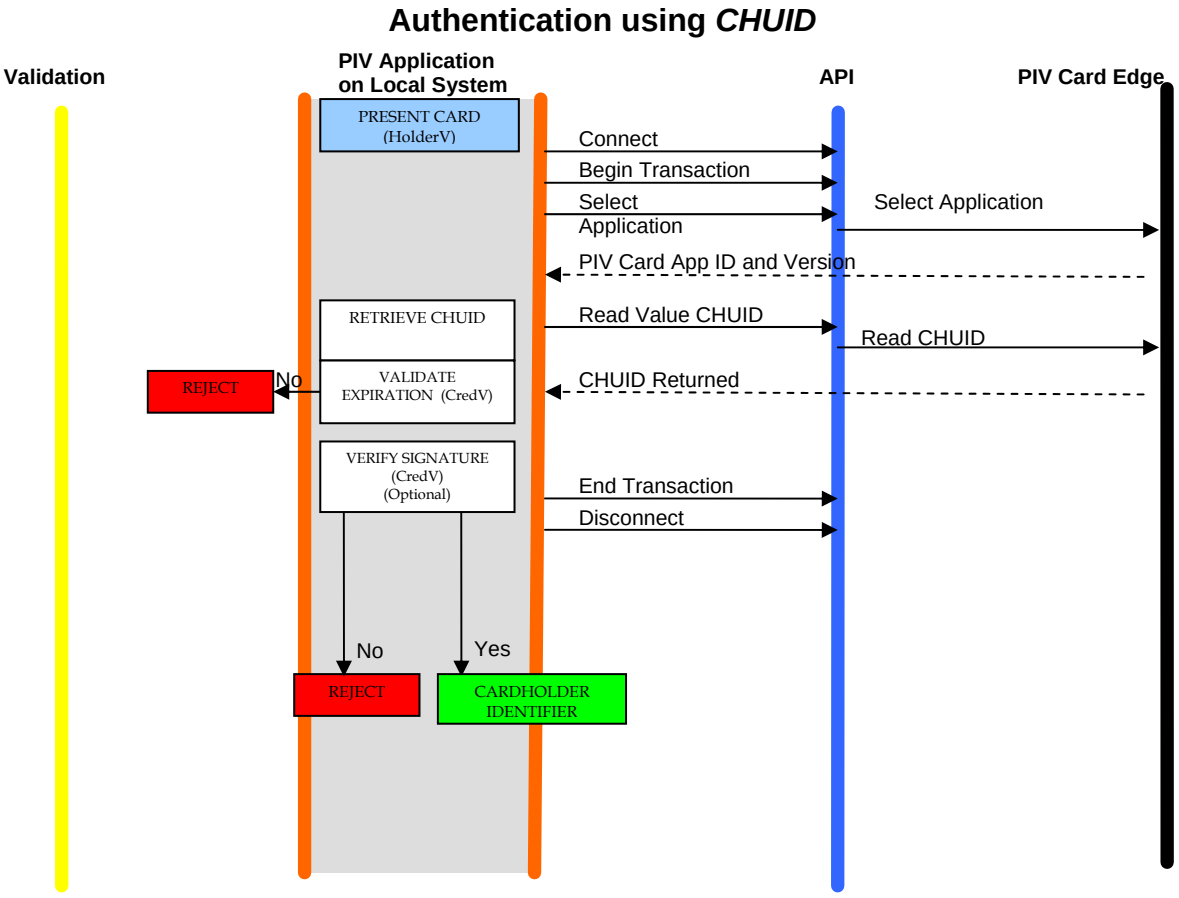

Figure B-2. Authentication using PIV *CHUID*

B.1.3 Authentication using PIV *Biometrics (BIO)*

The general authentication mechanism using the PIV biometrics is illustrated in Figure B-3.

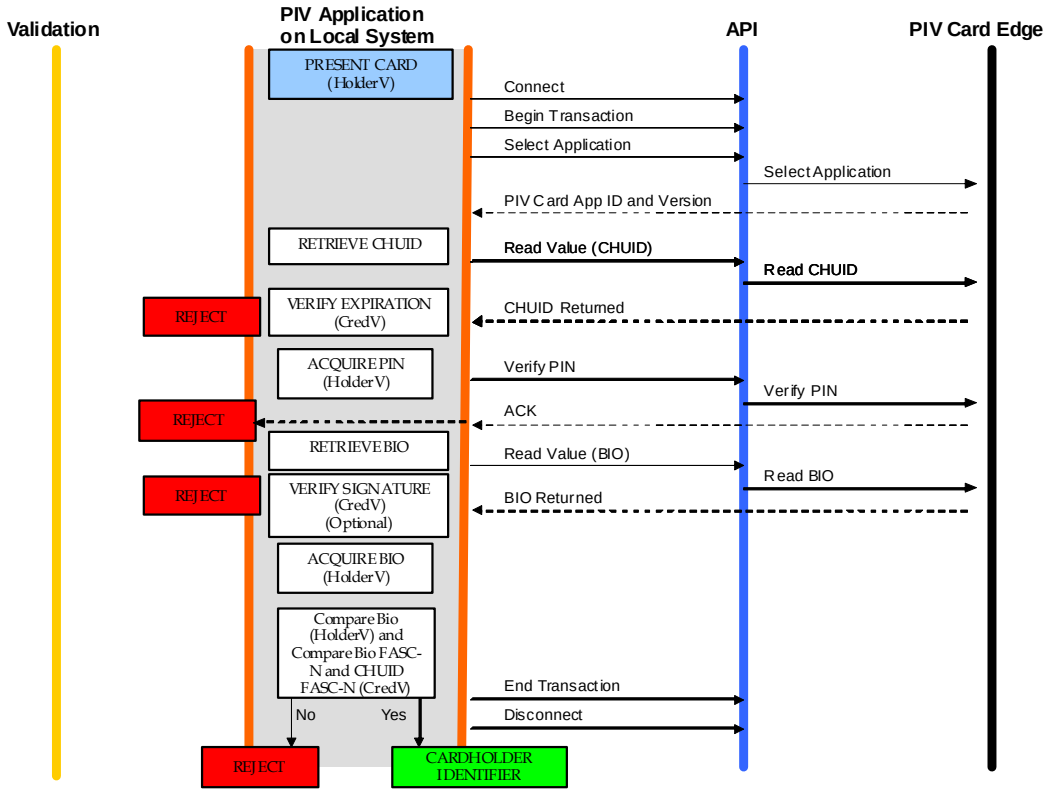

Figure B-3. Authentication using *PIV Biometrics (BIO)*

The assurance of authentication using the *PIV biometric* can be further increased if the live biometric sample is collected in an attended environment, with a human overseeing the process. The attended biometric authentication mechanism (BIO-A) is illustrated in Figure B-4.

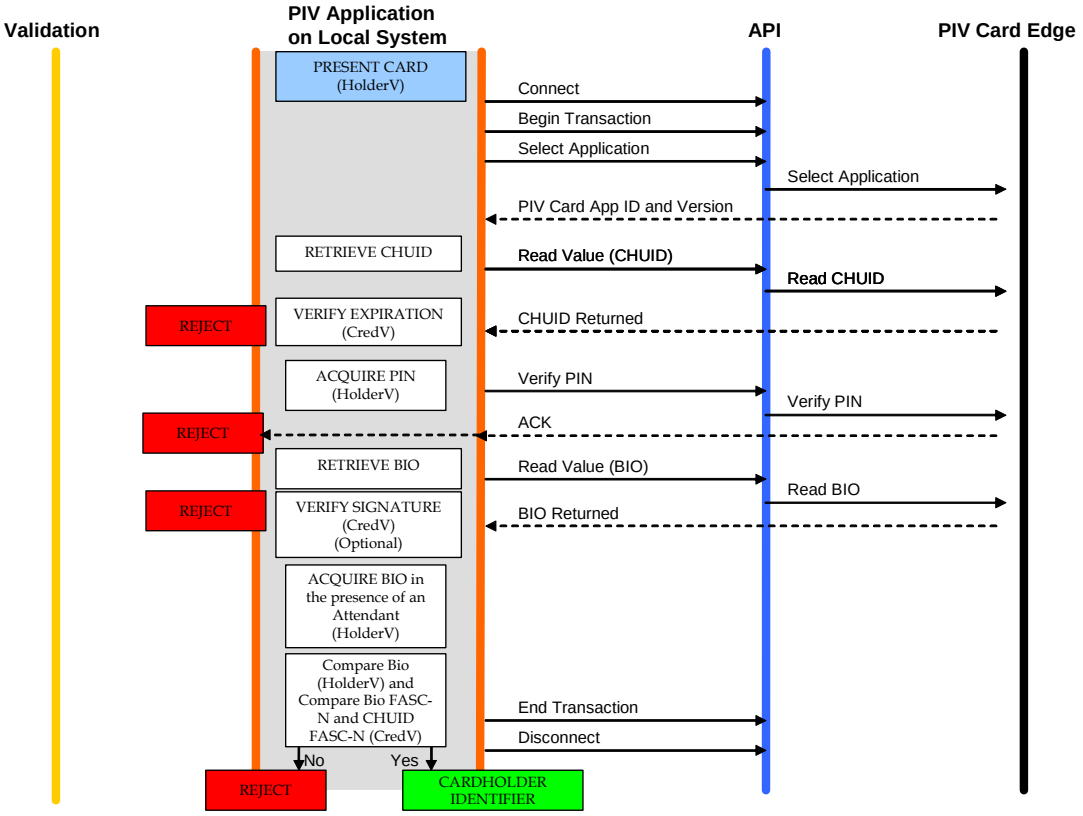

Figure B-4. Authentication using *PIV Biometrics Attended (BIO-A)*

B.1.4 Authentication using PIV Authentication Key

The authentication mechanism using the *PIV Authentication Key* is illustrated in Figure B-5.

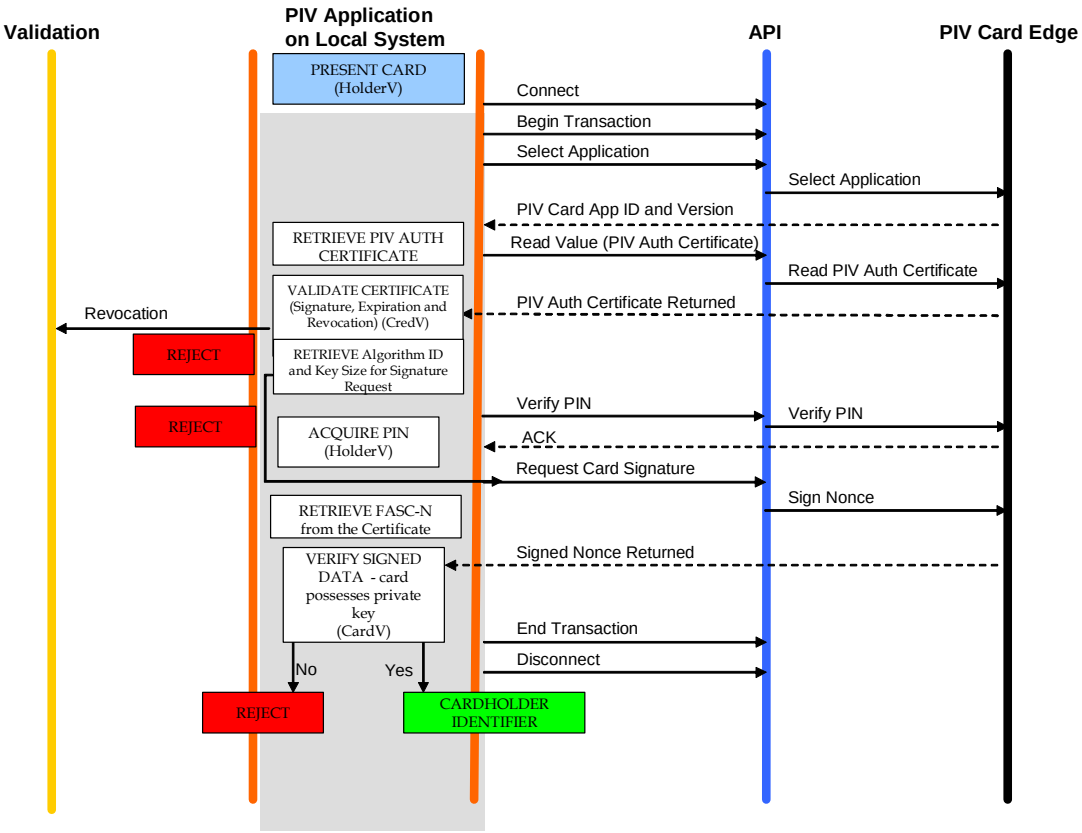

Figure B-5. Authentication using *PIV Authentication Key*

B.1.5 Authentication using Card Authentication Key

Authentication mechanisms using the *Card Authentication Key* are illustrated in Figures B-6 and B-7. Figure B-6 illustrates the use of an asymmetric *Card Authentication Key*, while figure B-7 uses a symmetric *Card Authentication Key* for the authentication mechanism. Both mechanisms provide "SOME" confidence in the assurance of the identity.

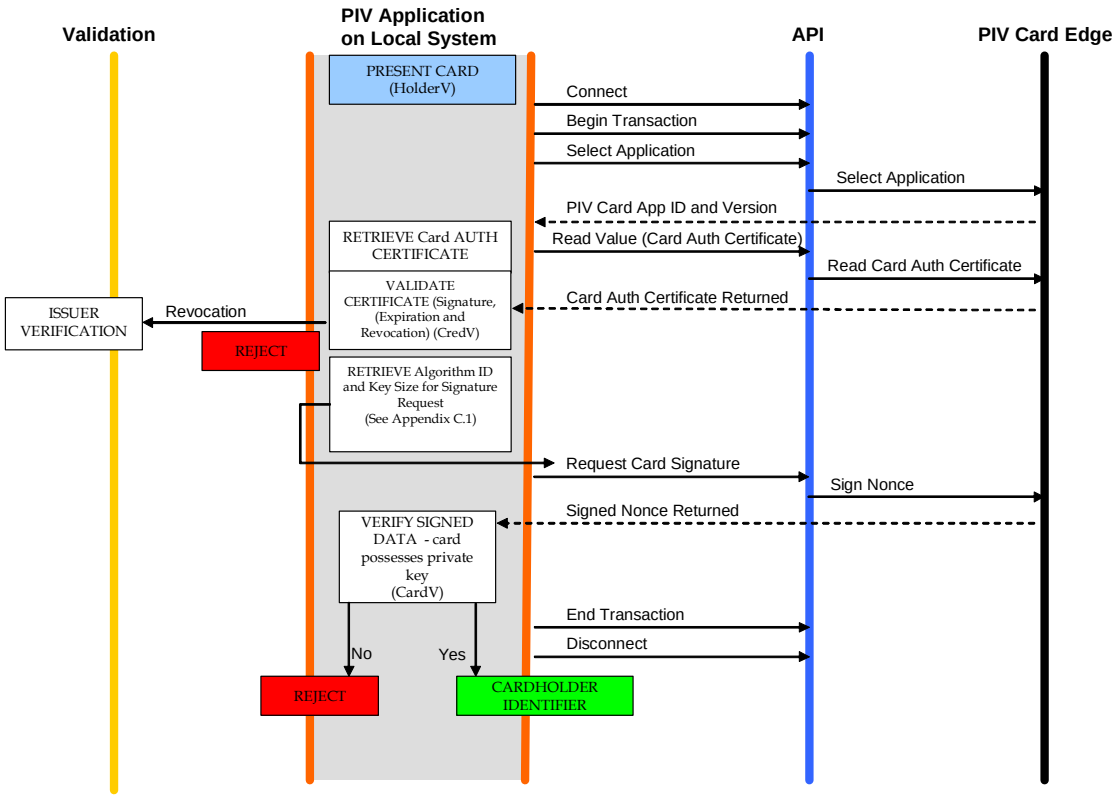

Figure B-6. Authentication using an asymmetric *Card Authentication Key*

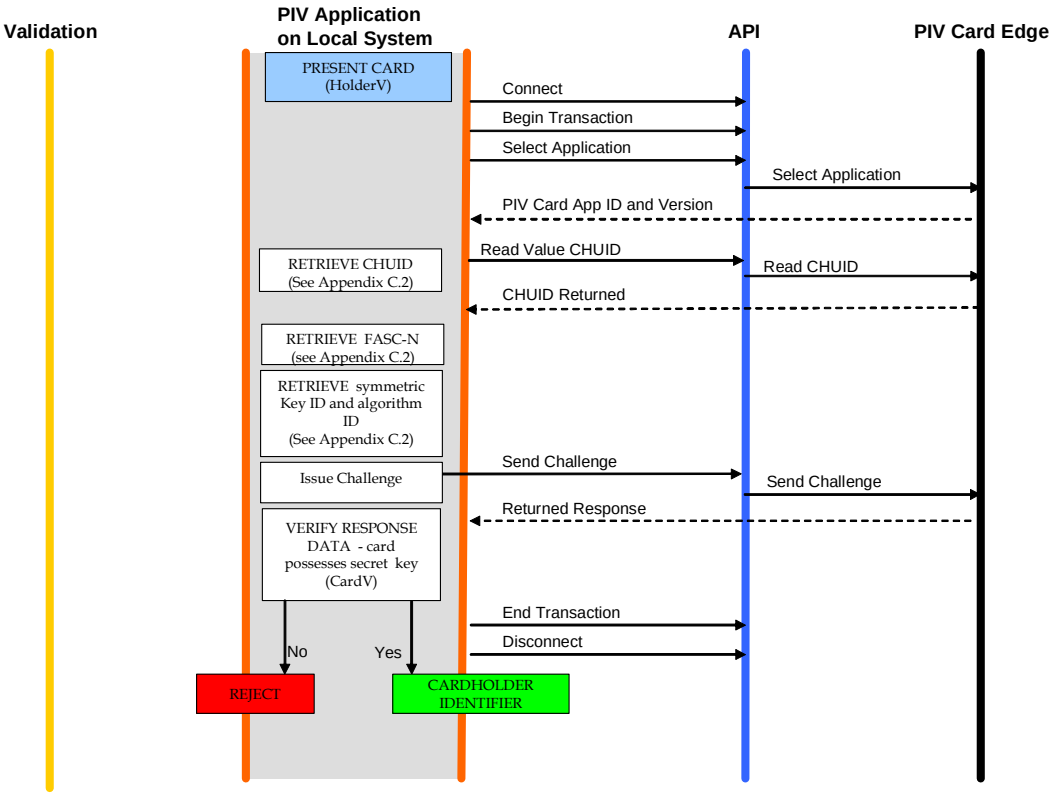

Figure B-7. Authentication using a symmetric *Card Authentication Key*

B.2 Summary Table

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

Table 40. Summary of PIV Authentication Mechanisms

PIV Authentication Mechanism	Card Validation Steps (CardV)	Credential Validation Steps (CredV)	Cardholder Validation Steps (HolderV)
PIV Visual Authentication	Counterfeit, tamper, and forgery check	Expiration check	Possession of Card Match of card visual characteristics with cardholder
PIV CHUID		Expiration check CHUID signature check (optional)	Possession of Card
Symmetric Card Authentication Key	Perform challenge and response with a PIV symmetric key		Possession of Card
Asymmetric Card Authentication Key	Perform challenge and response with a PIV asymmetric Card Authentication key, and validate signature on response	Card expiration check Certificate validation of a PIV certificate	Possession of Card
PIV Authentication Key	Perform challenge and response with a PIV asymmetric key, and validate signature on response	Card expiration check Certificate validation of a PIV certificate	Possession of Card Match PIN provided by Cardholder
PIV Biometric		Expiration check CHUID signature check (optional) PIV Bio signature check (optional) Match CHUID FASC-N with PIV Bio FASC-N	Possession of Card Match PIN provided by Cardholder Match Cardholder bio with PIV bio
PIV Biometric (Attended)		Expiration check CHUID signature check (optional) PIV Bio signature check (optional) Match CHUID FASC-N with PIV Bio FASC-N	Possession of Card Match PIN provided by Cardholder Match of Cardholder bio to PIV bio *in view of attendant*

Appendix C—PIV Algorithm Identifier Discovery

Relying Parties interact with many PIV Cards with the same native key-type implemented by different key sizes and algorithms.[14] For example, a relying party performing the authentication mechanism described in B.1.4 (Authentication using the *PIV Authentication Key*) can expect to perform a challenge and response cryptographic authentication with 1) a RSA 1024 bit key, 2) an RSA 2048 bit key, or 3) an elliptic curve key (P-256).

This appendix describes recommended procedures for key size and algorithm discovery (PIV algorithm ID discovery) to facilitate cryptographic authentication initiated by the relying party to make appropriate decisions for granting access to logical networks and systems as well as physical access control systems. The discovery procedure is defined in terms of asymmetric and symmetric cryptographic authentication.

C.1 PIV Algorithm Identifier Discovery for Asymmetric Cryptographic Authentication

As illustrated in the authentication mechanisms in Appendix B, an asymmetric cryptographic authentication involves issuing a challenge (request to sign a nonce) to the PIV Card. The relying party issuing the command provides the nonce to be signed, the key reference, and the PIV algorithm identifier as parameters of the command. The nonce is random data generated by the relying party and the key reference is known. The PIV algorithm identifier, on the other hand, is unknown to the relying party and needs to be identified in order to issue the challenge command. The PIV algorithm identifier can be derived from the previous steps of the authentication mechanism. The relying party, prior to the challenge command, retrieved and parsed the X.509 certificate from the card in order to 1) optionally validate the certificate and 2) extract the public key for the pending verification of the signed nonce once returned from the card. It is during the parsing of the X.509 certificate that the PIV algorithm identifier can be identified in two steps[15]:

Step 1: Algorithm Type Discovery:
The X.509 certificate stores the public key in the subjectPublicKeyInfo field. The subjectPublicKeyInfo data structure has an algorithm field, which includes an OID that identifies the public key's algorithm (RSA or ECC) as listed in Table 3-5 of SP 800-78.

Step 2: Key Size Discovery:
If the algorithm type, as determined in Step 1, is ECC then the key size is determined by the elliptic curve on which the key has been generated, which is P-256 for all elliptic curve PIV Authentication Keys and Card Authentication Keys.

If the algorithm type, as determined in Step 1, is RSA then the key size is determined by the public key's modulus. The public key appears in the subjectPublicKey field of subjectPublicKeyInfo and is encoded as a sequence that includes both the key's modulus and public exponent.

[14] Table 3-1, SP 800-78 lists the various PIV algorithm identifiers to choose one for each PIV key type.
[15] The PIV algorithm identifiers specify both the key size and the algorithm for the key references. Thus both values have to be discovered in order to derive the PIV algorithm identifier.

As a final step, the discovered X.509 algorithm OID and key size are mapped to the PIV Algorithm Identifiers as defined in Table 6-2 of SP 800-78. The relying party then proceeds to issue the general authenticate command to the card.

C.2 PIV Algorithm Identifier Discovery for Symmetric Cryptographic Authentication

In the absence of an X.509 certificate, as is the case with symmetric cryptography, the PIV algorithm identifier discovery mechanism has to rely on a lookup table residing at the local system. The table maps a unique card identifier and key reference (inputs) to an associated PIV algorithm identifier (output). The unique identifier supplied by the card shall be Agency Code || System Code || Credential Number of the FASC-N.

The optional *Card Authentication Key* can be a symmetric key or an asymmetric key. A relying party has no prior knowledge of 1) the key's existence or 2) the key's symmetric or asymmetric implementation. The following routine discovers the *Card Authentication Key's* native implementation:

1) Attempt to read the X.509 Certificate for Card Authentication.
   + If the first step succeeds, the *Card Authentication Key* is asymmetric. The asymmetric PIV algorithm identifier discovery mechanism (C.1) should be followed.
   + If the first step fails, the *Card Authentication Key* either does not exist or is a symmetric key.
2) Read the CHUID and extract the Agency Code || System code || Credential Number from the CHUID's FASC-N.
3) Attempt to retrieve the PIV algorithm identifier from the local lookup table.
   + If a valid PIV algorithm identifier is returned, the *Card Authentication Key* is symmetric.
   + If no algorithm identifier is returned, authentication cannot be performed using the *Card Authentication Key* either because the PIV Card does not implement the key or the local system cannot authenticate the response from the card.

Appendix D—Terms, Acronyms, and Notation

D.1 Terms

Algorithm Identifier	A PIV algorithm identifier is a one-byte identifier that specifies a cryptographic algorithm and key size. For symmetric cryptographic operations, the algorithm identifier also specifies a mode of operation (ECB).
Application Identifier	A globally unique identifier of a card application as defined in ISO/IEC 7816-4.
Application Session	The period of time within a card session between when a card application is selected and a different card application is selected or the card session ends.
Authenticatable Entity	An entity that can successfully participate in an authentication protocol with a card application.
BER-TLV Data Object	A data object coded according to ISO/IEC 8825-2.
Card	An integrated circuit card.
Card Application	A set of data objects and card commands that can be selected using an application identifier.
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 is specified a name, a description of logical content, a format, and a coding.
Interface Device	Synonym for card interface device.
Key Reference	A key reference is a one-byte identifier that specifies a cryptographic key according to its PIV Key Type. The identifier is part of the cryptographic material used in a cryptographic protocol, such as an authentication or a signing protocol.
MSCUID	An optional legacy identifier included for compatibility with Common Access Card and Government Smart Card Interoperability Specifications.
Object Identifier	A globally unique identifier of a data object as defined in ISO/IEC 8824-2.
PIV Key Type	The type of a key. The PIV Key Types are 1) PIV Authentication Key, 2) PIV Card Authentication Key, 3) PIV Digital Signature Key, 4) PIV Key Management Key, and 5) Card Application Administration Key.
Relying Party	An entity that relies upon the subscriber's credentials, typically to process a transaction or grant access to information or a system.

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.

D.2 Acronyms

ACR	Access Control Rule
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CAK	Card Authentication Key
CBEFF	Common Biometric Exchange Formats Framework
CCC	Card Capability Container
CHUID	Card Holder Unique Identifier
CMS	Cryptographic Message Syntax
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DG	Data Group
DTR	Derived Test Requirement
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FASC-N	Federal Agency Smart Credential Number
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GSC-IAB	Government Smart Card Interagency Advisory Board

GSC-IS	Government Smart Card Interoperability Specification
GUID	Global Unique Identification number
HSPD	Homeland Security Presidential Directive
ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
INCITS	InterNational Committee for Information Technology Standards
ISO	International Organization for Standardization
ITL	Information Technology Laboratory
LSB	Least Significant Bit
LRC	Longitudinal Redundancy Code
MRTD	Machine Readable Travel Document
MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
NPIVP	NIST Personal Identity Verification Program
OID	Object Identifier
OMB	Office of Management and Budget
PACS	Physical Access Control System
PIN	Personal Identification Number
PI	Person Identifier, a field in the FASC-N
PIV	Personal Identity Verification
PIV-C	PIV Compatible
PIV-I	PIV Interoperable
PIX	Proprietary Identifier Extension
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PUK	PIN Unblocking Key

RFU	Reserved for Future Use
RID	Registered application provider IDentifier
RSA	Rivest, Shamir, Aldeman
SCEPACS	Smart Card Enabled Physical Access Control System
SCP	ETSI Smart Card Project
SHA	Secure Hash Algorithm
SP	Special Publication
SW1	First byte of a two-byte status word
SW2	Second byte of a two-byte status word
TIG	Technical Implementation Guidance
TLV	Tag-Length-Value
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique IDentifier

D.3 Notation

The sixteen 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'. A sequence of bytes may be enclosed in single quotation marks, for example 'A0 00 00 01 16' rather than given as a sequence of individual bytes, 'A0' '00' '00' '01' '16'.

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

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

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

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

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

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

Appendix E—References

[1] Federal Information Processing Standard 201-1, Change Notice 1, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, March 2006. (See http://csrc.nist.gov)

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

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

[4] 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 Access Interagency Interoperability Working Group, July 30, 2004. (See http://fips201ep.cio.gov/documents/TIG_SCEPACS_v2.2.pdf)

[5] NIST Special Publication 800-87, *Codes for Identification of Federal and Federally-Assisted Organizations*, April 2008. (See http://csrc.nist.gov)

[6] *PKI for Machine Readable Travel Documents Offering ICC Read-Only Access Version - 1.1* Date - October 01, 2004. Published by authority of the Secretary General, International Civil Aviation Organization.

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

[8] 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).*

[9] NIST Special Publication 800-78-2, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification*, February 2010. (See http://csrc.nist.gov)

[10] *Personal Identity Verification Interoperability For Non-Federal Issuers*, May 2009. (See http://www.idmanagement.gov)

[11] IETF RFC 4122, "A Universally Unique IDentifier (UUID) URN Namespace," July 2005.

[12] IETF RFC 4530, "Lightweight Directory Access Protocol (LDAP) entryUUID Operational Attribute," June 2006.

[13] IETF RFC 5280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008.

[14] Federal Information Processing Standard 180-3, *Secure Hash Standard (SHS)*, October 2008. (See http://csrc.nist.gov)

NIST Special Publication 800-73-3

Interfaces for Personal Identity Verification – Part 2: End-Point PIV Card Application Card Command Interface

Ramaswamy Chandramouli
David Cooper
James F. Dray
Hildegard Ferraiolo
Scott B. Guthery
William MacGregor
Ketan Mehta

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

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

February 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

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

National Institute of Standards and Technology Special Publication 800-73-3, Part 2, 36 pages (February 2010)

Acknowledgements

The authors (Ramaswamy Chandramouli, David Cooper, James Dray, Hildegard Ferraiolo, William MacGregor of NIST, Ketan Mehta of Booz Allen Hamilton and Scott Guthery of HID Global) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Government Smart Card Interagency Advisory Board (GSC-IAB) and InterNational Committee for Information Technology Standards (INCITS) for providing detailed technical inputs to the SP 800-73 development process. The authors also gratefully acknowledge and appreciate the many contributions from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

Table of Contents

List of Appendices

List of Tables

1. Introduction

The Homeland Security Presidential Directive 12 (HSPD-12) called for a common identification standard to be adopted governing the interoperable use of identity credentials to allow physical and logical access to Federal government locations and systems. The Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201 (FIPS 201) [1] was developed to establish standards for identity credentials. Special Publication 800-73-3 (SP 800-73-3) specifies interface requirements for retrieving and using the identity credentials from the PIV Card[1] and is a companion document to FIPS 201.

1.1 Authority

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

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

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

1.2 Purpose

FIPS 201 defines procedures for the PIV lifecycle activities including identity proofing, registration, PIV Card issuance, and PIV Card usage. FIPS 201 also specifies that the identity credentials must be stored on a smart card. SP 800-73-3 contains technical specifications to interface with the smart 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, the specifications enumerate requirements where the standards include options and branches. SP 800-73-3 goes further by constraining implementers' interpretations of the normative standards. Such restrictions are designed to ease implementation, facilitate interoperability, and ensure performance, in a manner tailored for PIV applications.

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

1.3 Scope

SP 800-73-3 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 elaborated in Appendix B of SP 800-73-3 Part 1. Interoperability is defined as the use of PIV identity credentials such that client application programs, compliant card applications, and compliant integrated circuits cards (ICC) can be used interchangeably by all information processing systems across Federal agencies.

This Part, SP 800-73-3 Part 2 – *End-Point PIV Card Application Card Command Interface*, contains the technical specifications of the PIV Card command interface to the PIV Card. The specification defines the set of commands surfaced by the PIV Card Application at the card edge of the ICC.

1.4 Audience and Assumptions

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

Readers should also be aware of SP 800-73-3 Part 1, Section I, for the Revision History of SP800-73, Section II, which details Configuration Management Recommendations, and Section III, which specifies NPIVP Conformance Testing Procedures.

1.5 Content and Organization

All sections in this document are *normative* (i.e., mandatory for compliance) unless specified as *informative* (i.e., non-mandatory). Following is the structure of Part 2:

+ Section 1, *Introduction*, provides the purpose, scope, audience, and assumptions of the document and outlines its structure.
+ Section 2, *Overview: End-Point Concepts and Constructs*, describes the model of computation of the PIV Card Application and the PIV client application programming interface including information processing concepts and data representation constructs.
+ Section 3, *End-Point PIV Card Application Card Command Interface*, describes the set of commands accessible by the PIV Middleware to communicate with the PIV Card Application.
+ Appendix A, *Examples of the Use of GENERAL AUTHENTICATE*, demonstrates the GENERAL AUHTENTICATE command. This section is informative.
+ Appendix B, *Terms, Acronyms, and Notation*, contains the list of Terms and Acronyms used in this document and explains the notation in use. This section is informative.
+ Appendix C, *References*, contains the lists of documents used as references by this document. This section is informative.

2. Overview: End-Point Concepts and Constructs

SP 800-73-3 Part 2 and Part 3 define two interfaces to an ICC that contains the Personal Identity Verification Card Application: a low-level PIV Card Application card command interface (Part 2, card edge) and a high-level PIV client API (Part 3).

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

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

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

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

2.1 Unified Card Command Interface

The card command interface of the PIV Card Application is a unification of the two card command interfaces found in Government Smart Card Interoperability Specification (GSC-IS) [2].

This unification is accomplished by adopting the object-oriented model of computation of the GSC-IS virtual machine card edge and realizing its technical details using the data structures and operations found in the international ICC standards [3] underpinning the GSC-IS file system card edge. This brings the PIV Card Application into conformance with those standards with minimal impact on existing GSC-IS deployments.

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

2.1.1 Platform Requirements

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

+ global security status that includes the security status of a global cardholder PIN
+ application selection using a truncated Application Identifier (AID)

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

2.2 Namespaces of the PIV Card Application

AID, names, Tag-Length-Value (BER-TLV) [4] tags, ASN.1 [5] Object Identifiers (OIDs) and Proprietary Identifier eXtensions (PIXes) of the NIST Registered Application Provider IDentifier (RID) used on the PIV interfaces are specified in Part 1. Part 1 also specifies the use of all unspecified names, BER-TLV tags, OIDs, and values of algorithm identifiers, key references, and cryptographic mechanism identifiers.

2.3 Card Applications

Each command that appears on the card command interface shall be implemented by a *card application* that is resident on the ICC. The card command enables 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) [3, Part 4]. Except for the default applications, access to the card commands and data objects of a card application shall be gained by selecting the card application using its application identifier[2]. The PIX of the AID shall contain an encoding of the version of the card application. The AID of the Personal Identity Verification Card Application (PIV Card Application) is defined in Part 1.

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

2.3.1 Default Selected Card Application

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

2.4 Security Architecture

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

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

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

[2] Access to the default application, and its commands and objects, occurs immediately after a warm or cold card reset without an explicit SELECT command.

2.4.1 Access Control Rule

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

According to an access control rule, the action described by the access mode can be performed on the data object if and only if the security condition evaluates to TRUE for the current values of the security statuses. If there is no access control rule with an access mode describing a particular action, then that action shall never be performed on the data object.

2.4.2 Security Status

Associated with each authenticatable entity shall be a set of one or more Boolean variables each called a *security status indicator* of the authenticatable entity. Each security status indicator, in turn, is associated with a credential that can be used to authenticate the entity. The security status indicator of an authenticatable entity shall be TRUE if the credentials associated with the security status indicator of the authenticatable entity have been authenticated and FALSE otherwise.

A successful execution of an authentication protocol shall set the security status indicator associated with the credential used in the protocol to TRUE. An aborted or failed execution of an authentication protocol shall set the security status indicator associated with the credential used in the protocol to FALSE.

As an example, the credentials associated with two security status indicators of the cardholder might be: PIN and fingerprint. Demonstration of knowledge of the PIN is the authentication protocol for the first security status indicator wherein the PIN is the credential. Comparison of the fingerprint template on the card with a fingerprint acquired from the cardholder is the authentication protocol for the second security status indicator wherein the fingerprint is the credential. A security condition using these two security status indicators might be (PIN AND fingerprint).

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. In essence, when changing from one application to another, the global security status indicators shall remain unchanged.

A security status indicator is said to be an *application* security status indicator if it is set to FALSE when the currently selected application changes from one application to another. Every security status indicator is either a global security status indicator or an application security status indicator. The security status indicators associated with the PIV Card Application PIN, the PIN Unblocking Key (PUK), and the PIV Card Application Administrator are application security status indicators for the PIV Card Application, whereas the security status indicator associated with the Global PIN is a global security status indicator.

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

2.4.3 Authentication of an Individual

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

Personal identification numbers (PIV Card Application PINs and PUKs) presented to the card command interface shall be 8 bytes long. If the actual PIN length is less than 8 bytes it shall be padded to 8 bytes with 'FF'. The 'FF' padding bytes shall be appended to the actual PIN. The bytes comprising the PIV Card Application PIN shall be limited to values 0x30 – 0x39, the ASCII values for the decimal digits '0' – '9'. The bytes comprising the PUK shall be limited to the values 0x00 – 0xFE (i.e., shall not include 'FF'). For example,

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

If the Global PIN is used by the PIV Card Application then the above encoding, length, and padding requirements for the PIV Card Application PIN shall apply to the Global PIN.

2.5 Current State of the PIV Card Application

The elements of the *current state* of the PIV Card Application when the PIV Card Application is the currently selected application are described in Table 1.

Table 1. State of the PIV Card Application

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

3. End-Point PIV Card Application Card Command Interface

Table 2 lists the card commands surfaced by the PIV Card Application at the card edge of the ICC when it is the currently selected card application. All PIV Card Application card commands shall be supported by a PIV Card Application. Card commands indicated with a 'Yes' in the Command Chaining column shall support command chaining for transmitting a data string too long for a single command as defined in ISO/IEC 7816-4 [3].

Table 2. PIV Card Application Card Commands

Type	Name	Contact Interface	Contactless Interface	Security Condition for Use	Command Chaining
PIV Card Application Card Commands for Data Access	**SELECT**	Yes	Yes	Always	No
	GET DATA	Yes	Yes	Data Dependent. See Table 1, Part 1.	No
PIV Card Application Card Commands for Authentication	**VERIFY**	Yes	No	Always	No
	CHANGE REFERENCE DATA	Yes	No	PIN	No
	RESET RETRY COUNTER	Yes	No	PIN Unblocking Key	No
	GENERAL AUTHENTICATE	Yes	Yes (See Note)	Key Dependent. See Table 3, Part 1.	Yes
PIV Card Application Card Commands for Credential Initialization and Administration	**PUT DATA**	Yes	No	PIV Card Application Administrator	Yes
	GENERATE ASYMMETRIC KEY PAIR	Yes	No	PIV Card Application Administrator	Yes

The PIV Card Application shall return the status word of '6A81' (Function not supported) when it receives a card command on the contactless interface marked “No” in the Contactless Interface column in Table 2.

Note: Cryptographic protocols using private/secret keys requiring “PIN” security condition shall not be used on the contactless interface.

3.1 PIV Card Application Card Commands for Data Access

3.1.1 SELECT Card Command

The SELECT card command sets the currently selected application. The PIV Card Application shall be selected by providing its application identifier (see Part 1, Section 2.2) in the data field of the SELECT command.

There shall be at most one PIV Card Application on any ICC. The PIV Card Application can also be made the currently selected application by providing the right-truncated version (see Part 1, Section 2.2); that is, without the two-byte version number in the data field of the SELECT command.

The complete AID, including the two-byte version, of the PIV Card Application that became the currently selected card application upon successful execution of the SELECT command (using the full or right-truncated PIV AID) shall be returned in the application property template.

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

If the currently selected application is the PIV Card Application when the SELECT command is given and the AID in the data field of the SELECT command is not the PIV Card Application (nor the right-truncated version thereof), but a valid AID supported by the ICC, then the PIV Card Application shall be deselected and all the PIV Card Application security status indicators in the PIV Card Application shall be set to FALSE.

If the currently selected application is the PIV Card Application when the SELECT command is given and the AID in the data field of the SELECT command is an invalid AID not supported by the ICC, then the PIV Card Application shall remain the current selected card application and all PIV Card Application security status indicators shall remain unchanged.

Command Syntax

CLA	'00'
INS	'A4'
P1	'04'
P2	'00'
L_c	Length of application identifier
Data Field	AID of the PIV Card Application using the full AID or by providing the right-truncated AID (See Section 2.2, Part 1)
L_e	Length of application property template

Response Syntax

Data Field	Application property template (APT). See Table 3 below
SW1-SW2	Status word

Upon selection, the PIV Card Application shall return the application property template described in Table 3.

Table 3. Data Objects in the PIV Card Application Property Template (Tag '61')

Description	Tag	M/O	Comment
Application identifier of application	'4F'	M	The PIX of the AID includes the encoding of the version of the PIV Card Application. See Section 2.2, Part 1.
Coexistent tag allocation authority	'79'	M	Coexistent tag allocation authority template. See Table 4.
Application label	'50'	O	Text describing the application; e.g., for use on a man-machine interface.
Uniform resource locator	'5F50'	O	Reference to the specification describing the application.

Table 4. Data Objects in a Coexistent Tag Allocation Authority Template (Tag '79')

Description	Tag	M/O	Comment
Application identifier	'4F'	M	See Section 2.2, Part 1

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

3.1.2 GET DATA Card Command

The GET DATA card command retrieves the data content of the single data object whose tag is given in the data field.[3]

Command Syntax

CLA	'00'
INS	'CB'
P1	'3F'
P2	'FF'
L_c	Length of data field*
Data Field	See Table 5
L_e	Number of data content bytes to be retrieved.

* The L_c value is '05' for all PIV data objects except for the 0x7E interindustry tag (Discovery Object) and the Application Property Template (APT), which have an L_c value of '03'.

[3] It is assumed that the GET DATA command will use the GET RESPONSE command to accomplish the reading of larger PIV Data Objects. The GET RESPONSE command is illustrated in A.3.1 (Command 3).

Table 5. Data Objects in the Data Field of the GET DATA Card Command

Name	Tag	M/O	Comment
Tag list	'5C'	M	BER-TLV tag of the data object to be retrieved. See Table 2, Part 1.

Response Syntax

For the (optional) 0x7E Discovery Object (if present):

Data Field	BER-TLV of the 0x7E Discovery data object (see Section 3.2.6, Part 1 for an example of the Discovery Object's structure returned in the data field).
SW1-SW2	Status word

For all other PIV data objects:

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

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

3.2 PIV Card Application Card Commands for Authentication

3.2.1 VERIFY Card Command

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

Key reference '80' specific to the PIV Card Application (i.e., local key references) and, optionally, the Global PIN with key reference '00' are the only key references that may be verified by the PIV Card Application's VERIFY command.

Key reference '80' shall be able to be verified by the PIV Card Application VERIFY command.

If the PIV Card Application contains the Discovery Object as described in Part 1, and the first byte of the PIN Usage Policy value is 0x60, then key reference '00' shall be able to be verified by the PIV Card Application VERIFY command.

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

If the authentication data in the command data field does not satisfy the criteria in Section 2.4.3, then the card command shall fail, and the PIV Card Application shall return the status word '6A 80'.

If the authentication data in the command data field does not match reference data associated with the key reference, then the card command shall fail.

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

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

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

Command Syntax

CLA	'00'
INS	'20'
P1	'00'
P2	Key reference. See Part 1, Table 3.
L_c	'00'[4] or '08'
Data Field	Absent[4] or PIN reference data as described in Section 2.4.3.
L_e	Empty

Response Syntax

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

3.2.2 CHANGE REFERENCE DATA Card Command

The CHANGE REFERENCE DATA card command initiates the comparison of the verification data with the current value of the reference data and if this comparison is successful, replaces the reference data with new reference data.

[4] If L_c=0x00 and the command data field is empty, the command can be used to retrieve the number of further retries allowed ('63 CX'), or to check whether verification is not needed ('90 00').

Only reference data associated with key references '80' and '81' specific to the PIV Card Application (i.e., local key reference) and the Global PIN with key reference '00' may be changed by the PIV Card Application CHANGE REFERENCE DATA command.

Key reference '80' reference data shall be changed by the PIV Card Application CHANGE REFERENCE DATA command. The ability to change reference data associated with key references '81' and '00' using the PIV Card Application CHANGE REFERENCE DATA command is optional.

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

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

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

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

If either the current reference data or the new reference data in the command data field of the command does not satisfy the criteria in Section 2.4.3, the PIV Card Application shall not change the reference data associated with the key reference and shall return the status word '6A 80'.

Command Syntax

CLA	'00'
INS	'24'
P1	'00'
P2	Key reference. See Part 1, Table 3
L_c	'10'
Data Field	Current PIN reference data concatenated without delimitation with the new PIN reference data, both PINs as described in Section 2.4.3
L_e	Empty

Response Syntax

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

3.2.3 RESET RETRY COUNTER Card Command

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

The only key reference allowed in the P1 parameter of the RESET RETRY COUNTER command is the PIV Card Application PIN.

If the current value of the PUK's retry counter is zero, then the PIN's retry counter shall not be reset, and the PIV Card Application shall return the status word '69 83'.

If the card command succeeds, then the PIN's retry counter shall be set to its reset retry value. Optionally, the PUK's retry counter may be set to its initial reset retry value. The security status of the PIN's key reference shall not be changed.

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

The initial retry counter associated with the PUK; i.e., the number of failures of the RESET RETRY COUNTER command before the PUK's retry counter reaches zero, is issuer dependent.

If the reset retry counter reference data (PUK) or the new reference data (PIN) in the command field of the command does not satisfy the criteria in Section 2.4.3, the PIV Card Application shall not reset the retry counter associated with the PIN and shall return the status word '6A 80'.

Command Syntax

CLA	'00'
INS	'2C'
P1	'00'
P2	Key reference '80'. See Part 1, Table 3
L_c	'10'
Data Field	Reset retry counter reference data (PUK) concatenated without delimitation with the new reference data (PIN) (both PUK and PIN as described in Section 2.4.3)
L_e	Empty

Response Syntax

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

3.2.4 GENERAL AUTHENTICATE Card Command

The GENERAL AUTHENTICATE card command performs a cryptographic operation such as an authentication protocol using the data provided in the data field of the command and returns the result of the cryptographic operation in the response data field.[5]

The GENERAL AUTHENTICATE command shall be used with the PIV authentication Keys ('9A', '9B', '9E') to authenticate the card or a card application to the client application (INTERNAL AUTHENTICATE), to authenticate an entity to the card (EXTERNAL AUTHENTICATE), and to perform a mutual authentication between the card and an entity external to the card (MUTUAL AUTHENTICATE).

The GENERAL AUTHENTICATE command shall be used with the PIV Digital Signature Key ('9C') to realize the signing functionality on the PIV client application programming interface. Data to be signed is expected to be hashed off card. Appendix A, Section A.3 illustrates the use of the GENERAL AUTHENTICATE command for signature generation.

The GENERAL AUTHENTICATE command shall be used with the PIV Key Management Key ('9D') and the retired PIV Key Management Keys ('82' – '95') to realize key establishment schemes specified in SP 800-78 (ECDH and RSA). Appendix A.4 illustrates the use of the GENERAL AUTHENTICATE command for key establishment schemes aided by the PIV Card Application.

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

Command Syntax

CLA	'00' or '10' indicating command chaining
INS	'87'
P1	Algorithm reference. See Table 6-2, SP 800-78 [6]
P2	Key reference: • See Table 3, Part 1 for key references of retired private Key Management Keys • See Table 6-1, SP 800-78 for all other key references
L_c	Length of data field
Data Field	See Table 6
L_e	Absent or length of expected response

[5] For cryptographic operations with larger keys, e.g., RSA 2048, it is assumed that the GENERAL AUTHENTICATE command will use the GET RESPONSE command to return the complete result of the cryptographic operation. The GET RESPONSE command is illustrated in A.3.1 (Command 3).

Table 6. Data Objects in the Dynamic Authentication Template (Tag '7C')

Name	Tag	M/O	Description
Witness	'80'	C	Demonstration of knowledge of a fact without revealing the fact. An empty witness is a request for a witness.
Challenge	'81'	C	One or more random numbers or byte sequences to be used in the authentication protocol.
Response	'82'	C	A sequence of bytes encoding a response step in an authentication protocol.
Exponentiation	'85'	C	A parameter used in ECDH key agreement protocol.

The data objects that appear in the dynamic authentication template (tag '7C') in the data field of the GENERAL AUTHENTICATE card command depend on the authentication protocol being executed. The Witness ('80') contains encrypted data (unrevealed fact). This data is decrypted by the card. The Challenge ('81') contains clear data (byte sequence), which is encrypted by the card. The Response (tag '82') contains either the decrypted data from tag '80' or the encrypted data from tag '81'. Note that the empty tags (i.e., tags with no data) return the same tag with content (they can be seen as “requests for requests”):

+ '80 00' Returns '80 TL <encrypted random>' (as per definition)
+ '81 00' Returns '81 TL <random>' (as per external auth example)

Response Syntax

Data Field	Absent, authentication-related data, signed data, shared secret, or transported key
SW1-SW2	Status word

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

3.3 PIV Card Application Card Commands for Credential Initialization and Administration

3.3.1 PUT DATA Card Command

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

Command Syntax

CLA	'00' or '10' indicating command chaining
INS	'DB'
P1	'3F'
P2	'FF'
L_c	Length of data field
Data Field	See Tables 7 and 8
L_e	Empty

Table 7. Data Objects in the Data Field of the PUT DATA Card Command for the Discovery Object

For the 0x7E Discovery Object (if present):

Tag	M/O	Description
'7E'	M	BER-TLV of tag '7E' as illustrated in Section 3.2.6, Part 1

Table 8. Data Objects in the Data Field of the PUT DATA Card Command for all other PIV Data Objects

For all other PIV Data objects:

Name	Tag	M/O	Description
Tag list	'5C'	M	Tag of the data object whose data content is to be replaced. See Table 2, Part 1.
Data	'53'	M	Data with tag '53' as an unstructured byte sequence.

Response Syntax

Data Field	Absent
SW1-SW2	Status word

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

3.3.2 GENERATE ASYMMETRIC KEY PAIR Card Command

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

Command Syntax

CLA	'00' or '10' indicating command chaining.
INS	'47'
P1	'00'
P2	See SP 800-78 Table 6-1 for a list of the PIV Key References
L_c	Length of data field
Data Field	Control reference template. See Table 9
L_e	Length of public key of data object template

Table 9. Data Objects in the Template (Tag 'AC')

Name	Tag	M/O	Description
Cryptographic mechanism identifier	'80'	M	See Part 1, Table 4
Parameter	'81'	C	Specific to the cryptographic mechanism

Response Syntax

Data Field	Data objects of public key of generated key pair. See Table 10
SW1-SW2	Status word

Table 10. Data Objects in the Template (Tag '7F49')

Name	Tag
Public key data objects for RSA	
Modulus	'81'
Public exponent	'82'
Public key data objects for ECDSA	
Point	'86'

The public key data object in tag '86' is encoded as follows:

Tag	Length	Value
86	L	04 || X || Y (see Section 2.3.3 of [8])

Note: The octet '04' indicates that the X and Y coordinates of point P are encoded without the use of point compression. The length L is 65 bytes for points on Curve P-256 and 97 bytes for points on Curve P-384.

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

Appendix A—Examples of the Use of GENERAL AUTHENTICATE

A.1 Authentication of the PIV Card Application Administrator

The PIV Card Application Administrator is authenticated by the PIV Card Application using a challenge/response protocol. A challenge retrieved from the PIV Card Application is encrypted by the client application and returned to the PIV Card Application associated with key reference '9B', the key reference to the Card Management Key[6]. The PIV Card Application decrypts the response using this reference data and the algorithm associated with the key reference (for example 3 Key Triple DES – ECB, algorithm identifier '00'). If this decrypted value matches the previously provided challenge, then the security status indicator of the PIV Card Application Administrator is set to TRUE within the PIV Card Application.

Table 11 shows the GENERAL AUTHENTICATE card commands sent to the PIV Card Application to realize this particular challenge/response protocol.

Table 11. Authentication of PIV Card Application Administrator

Command	Response	Comment
'00 87 00 9B 04 7C 02 81 00'		Client application requests a challenge from the PIV Card Application
	'7C 0A 81 08 01 02 03 04 05 06 07 08'	Challenge returned to client application by the PIV Card Application
'00 87 00 9B 0C 7C 0A 82 08 88 77 66 55 44 33 22 11'		Client application returns the encryption of the challenge ('88 77 66 55 44 33 22 11') referencing algorithm '00' and key reference '9B'. See Tables 6-1 and 6-2 of SP 800-78.
	'90 00'	PIV Card Application indicates successful authentication of PIV Card Application Administrator after decrypting '88 77 66 55 44 33 22 11' using the referenced algorithm and key and getting '01 02 03 04 05 06 07 08'.

A.2 Validation of the PIV Card Application

The PIV Card Application is validated by first retrieving the X.509 Certificate of the PIV Authentication Key (OID 2.16.840.1.101.3.7.2.1.1) and validating the certificate. Assuming the certificate is valid, the client application requests the PIV Card Application to sign a challenge using the private key associated with this certificate (i.e., key reference '9A') and the appropriate algorithm (e.g., algorithm identifier '06'), which can be determined from the certificate as described in Part 1, Appendix C.1. The response is

[6] The Card Management Key is the PIV Card Application Administration Key used for managing the PIV Card Application.

verified using the public key in the certificate. If the signature verifies, then the PIV Card Application is validated.

Table 12 shows the GENERAL AUTHENTICATE card commands sent to the PIV Card Application to realize the validation of the PIV Card Application when the X.509 Certificate for PIV Authentication includes a 1024-bit RSA public key.

Table 12. Validation of the PIV Card Application Using GENERAL AUTHENTICATE

Command	Response	Comment
'00 87 06 9A 88 7C 81 85 82 00 81 81 80 00 01 FF … 58 EA C3 00' ("…" represents 122 bytes of challenge data)		Client application sends a challenge to the PIV Card Application indicating the reference data associated with key reference '9A' is to be used with algorithm '06'. See Tables 6-1 and 6-2 in SP 800-78. The challenge data, which in this example is encoded in accordance with the PKCS #1 v.1.5 signature padding scheme, is '00 01 FF ... 58 EA C3'. L_e is '00' to indicate that the expected length of the response data field is 256 bytes.
	'7C 81 83 82 81 80 88 0E BA … 41 E6 EE 90 00' ("…" represents 122 bytes of the signed challenge)	PIV Card Application returns the result of signing the challenge using the indicated key reference data and algorithm ('88 0E BA … 41 E6 EE').

A.3 Signature Generation with the Digital Signature Key

The GENERAL AUTHENTICATE command can be used to generate signatures. The pre-signature hash and padding (if applicable) is computed off card. The PIV Card Application receives the hashed value of the original message, applies the private signature key (key reference '9C'), and returns the resulting signature to the client application.

Listed below are the card commands sent to the PIV Card Application to generate a signature. It is assumed that the cardholder PIN has been successfully verified prior to sending the GENERAL AUTHENTICATE command.

A.3.1 RSA

This example illustrates signature generation using RSA 2048 (i.e., algorithm identifier '07'). Command chaining is used in the first command since the padded hash value sent to the card for signature generation is bigger than the length of the data field.

Command 1: (GENERAL AUTHENTICATE – first chain):

CLA	'10' indicating command chaining
INS	'87'
P1	'07'
P2	'9C'
L_c	Length of data field
Data Field	'7C' – L1 { '82' '00' '81' L2 {first part of the PKCS #1 v1.5 or PSS padded message hash value }}
L_e	Absent (no response expected)

Response 1:

Data Field	Absent
SW1-SW2	'90 00' (Status word)

Command 2: (GENERAL AUTHENTICATE – last chain):

CLA	'00' indicates last command of the chain.
INS	'87'
P1	'07'
P2	'9C'
L_c	Length of data field
Data Field	{second and last part of the PKCS #1 v1.5 or PSS padded message hash value}
L_e	Length of expected response

Response 2:

Data Field	'7C' – L1 {'82' L2 {first part of signature} }
SW1-SW2	'61 xx' where xx indicates the number of bytes remaining to send by the PIV Card Application

Command 3: (GET RESPONSE APDU):

CLA	'00'
INS	'C0'
P1	'00'
P2	'00'
L_e	xx Length of remaining response as indicated by previous SW1-SW2

Response 3:

Data Field	{second and last part of signature}
SW1-SW2	'90 00' (Status word)

A.3.2 ECDSA

The following example illustrates signature generation with ECDSA using ECC: Curve P-256 (i.e., algorithm identifier '11'). Command chaining is not used in this example, as the hash value fits into the data field of the command. Padding does not apply to ECDSA.

Command – GENERAL AUTHENTICATE

CLA	'00'
INS	'87'
P1	'11'
P2	'9C'
L_c	Length of data field
Data Field	'7C' – L1 { '82' '00' '81' L2 {hash value of message}}
L_e	Length of expected response

Response:

Data Field	'7C' – L1 {'82' L2 (r,s)} where • (r,s) is DER encoded with the following ASN.1 structure: Ecdsa-Sig-Value ::= SEQUENCE { r INTEGER, s INTEGER } • L1 is the length of tag '82' TLV structure • L2 is the length of the DER encoded Ecdsa-Sig-Value structure
SW1-SW2	'90 00' (Status word)

A.4 Key Establishment Schemes with the PIV Key Management Key

FIPS 201 specifies an optional public key pair and associated X.509 Certificate for Key Management. The Key Management Key (KMK) is further defined in SP 800-78, which defines two distinct key establishment schemes for the KMK:

1) RSA key transport and
2) Elliptic Curve Diffie-Hellman (ECDH) key agreement.

The use of the KMK for RSA key transport and ECDH key agreement is discussed in Sections A.4.1 and A.4.2, respectively.

A.4.1 RSA Key Transport

In general, RSA transport keys are used to establish symmetric keys, where a sender encrypts a symmetric key with the receiver's public key and sends the encrypted key to the receiver. The receiver decrypts the encrypted key with the corresponding private key. The decrypted symmetric key subsequently is used by both parties to protect further communication between them. Many types of security protocols employ

the RSA key transport technique. S/MIME for secure email and TLS for secure web communications are two of the many protocols employing RSA transport keys to distribute symmetric keys between entities.

A.4.1.1. RSA Key Transport with the PIV KMK

As specified in SP 800-78, the on-card private KMK can be an RSA transport key that complies with PKCS #1[9]. In the scenario described above, a sender encrypts a symmetric key with the KMK's public RSA transport key. The role of the on-card KMK private RSA transport key is to decrypt the sender's symmetric key on behalf of the cardholder and provide it to the client application cryptographic module.

A.4.1.1.1 The GENERAL AUTHENTICATE Command

Listed below are the card commands sent to the PIV Card to decrypt the symmetric key. It is assumed that the cardholder's PIN has been successfully verified prior to sending the GENERAL AUTHENTICATE command to the card.

Command 1 – GENERAL AUTHENTICATE (first chain)

CLA	'10' indicates command chaining
INS	'87'
P1	'07'
P2	'9D'
L_c	Length of data field
Data Field	'7C' – L1 {'82' '00' '81' L2 {first part of c}} where • c is the cipher text representative as defined in Section 5.1.2 of PKCS #1 v2.1
L_e	Absent (no response expected)

Response 1:

Data Field	Absent
SW1-SW2	'90 00' (Status word)

Command 2 – GENERAL AUTHENTICATE (last chain)

CLA	'00' indicates last command of the chain
INS	'87'
P1	'07'
P2	'9D'
L_c	Length of data field
Data Field	{second and last part of ciphertext representative c}}
L_e	Length of expected response

Response 2:

Data Field	'7C' – L1 {'82' L2 {first part of message representative m}} where m is as defined in PKCS #1 v2.1 [9] Section 5.1.2

SW1-SW2	'61 xx' where x indicates the number of bytes remaining to send

Command 3: (GET RESPONSE APDU):

CLA	'00'
INS	'C0'
P1	'00'
P2	'00'
L_e	xx Length of remaining response as indicated by previous SW1-SW2

Response 3:

Data Field	{second and last part of message representative m}
SW1-SW2	'90 00' (Status word)

A.4.2 Elliptic Curve Cryptography Diffie-Hellman

An ECDH key agreement scheme does not send an encrypted symmetric key to the participating entities. Instead, the two entities involved in the key agreement scheme compute a shared secret by combining their ECC private key(s) with the other party's public key(s). The resulting shared secret (Z) serves as an input to a Key Derivation Function (KDF), which each entity independently invokes to derive a common secret key. The secret key may be used as a session key or may be used to encrypt a session key.

A. 4.2.1 ECDH with the PIV KMK

The PIV Card supports ECDH key agreement by performing the Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC CDH) Primitive (see Section 5.7.1.2 of SP 800-56A [7]) using its ECC KMK private key and an ECC public key that is provided as input to the GENERAL AUTHENTICATE command. All other procedures required to complete the key agreement are performed by the cardholder's client application and its associated cryptographic module.

A.4.2.1.1 The GENERAL AUTHENTICATE Command

The sequence of commands to perform the ECC CDH Primitive from Section 5.7.1.2 of SP 800-56A with the private ECC KMK is illustrated below for ECC: Curve P-256:

Command – GENERAL AUTHENTICATE

CLA	'00'
INS	'87'
P1	'11'
P2	'9D'
L_c	Length of data field
Data Field	'7C' – L1 {'82' '00' '85' L2 { '04' || X || Y}} , where • '04 || X || Y' is the other party's public key, a point on Curve P-256, encoded without the use of point compression as described in Section 2.3.3 of [8].

	• The length of each coordinate (X and Y) is 32 bytes and • The value of L2 is 65 bytes
L_e	Length of expected response

Response:

Data Field	'7C' – L1 {'82' L2 {shared secret Z}} where • Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2 • L2 is 32 bytes
SW1-SW2	'90 00' (Status word)

A.4.2.2 PIV KMK Specific ECDH Key Agreement Schemes

SP 800-56A describes five different ECDH key agreement schemes that a client application cryptographic module may implement. These schemes differ in 1) the number of keys (1 or 2) and 2) the type of keys (ephemeral or static) used by each party. Since the PIV Card only computes the ECC CDH Primitive using its static private key, the client application cryptographic module only employs the PIV Card in implementing an ECDH key agreement scheme when the scheme involves the use of the cardholder's static key pair. The ECDH key agreement schemes that involve the use of at least one party's static key pair, and thus may involve the use of the PIV Card are:

+ C(2, 2) – Each party has a static key pair and generates an ephemeral key pair (see Section 6.1.1 of SP 800-56A)

  In this scheme, the information sent between the client application and the PIV Card is the same when acting as the initiator or the responder; the other party's static public key is sent to the PIV Card; and a static shared secret is returned by the PIV Card in plaintext. Note that an ephemeral key pair is generated by the client application, and the private key of that key pair is combined with the other party's ephemeral public key to produce an ephemeral shared secret.

+ C(1, 2) – The initiator has a static key pair and generates an ephemeral key pair, while the responder has a static key pair (see Section 6.2.1 of SP 800-56A)

  When the cardholder is acting as the initiator, the other party's static public key is sent to the PIV Card; and a static shared secret is returned in plaintext by the PIV Card. Note that in this case, an ephemeral key pair is generated by the client application cryptographic module, and the corresponding ephemeral private key is combined with the other party's static public key to produce a second shared secret.

  When the cardholder is acting as the responder, two public keys are sent by the client application to the PIV Card (the other party's static and ephemeral public keys), and two shared secrets are returned in plaintext (the static shared secret and the ephemeral shared secret). Note that two GENERAL AUTHENTICATE commands are required to provide the two shared secrets to the client application's cryptographic module.

+ C(1,1) – The initiator generates only an ephemeral key pair, while the responder has only a static key pair (see Section 6.2.2 of SP 800-56A).

In this scheme, the PIV Card is only employed by the client application if the cardholder is acting as the responder. In this case, the other party's ephemeral public key is sent to the PIV Card, and the shared secret is returned by the PIV Card in plaintext.

+ C(0,2) – Both the initiator and responder use only static key pairs (see Section 6.3 of SP 800-56A)

  In the C(0,2) scheme, the information sent between the client application's cryptographic module and the PIV Card is the same when acting as the initiator or the responder; the other party's static public key is sent to the PIV Card, and the static shared secret is returned in plaintext. Note that for this scheme, the client application cryptographic module also generates a nonce when acting as the initiator of the scheme.

The C(2,0) scheme does not involve the use of static keys and so the PIV Card would not be involved in the implementation of this scheme.

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.
Algorithm Identifier	A PIV algorithm identifier is a one-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).
Authenticatable Entity	An entity that can successfully participate in an authentication protocol with a card application.
BER-TLV Data Object	A data object coded according to ISO/IEC 8825-2.
Card	An integrated circuit card.
Card Application	A set of data objects and card commands that can be selected using an application identifier.
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.
Key Reference	A PIV key reference is a one-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 defined in ISO/IEC 8824-2.
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 signify the success of or errors encountered during said processing.
Template	A (constructed) BER-TLV data object whose value field contains specific BER-TLV data objects.

B.2 Acronyms

AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
APT	Application Property Template
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CLA	Class (first) byte of a card command
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
EC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GSC-IAB	Government Smart Card Interagency Advisory Board
GSC-IS	Government Smart Card Interoperability Specification
HSPD	Homeland Security Presidential Directive
ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
INS	Instruction (second) byte of a card command
INCITS	InterNational Committee for Information Technology Standards
ISO	International Organization for Standardization

ITL	Information Technology Laboratory
KDF	Key Derivation Function
LSB	Least Significant Bit
MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
OID	Object Identifier
OMB	Office of Management and Budget
P1	First parameter of a card command
P2	Second parameter of a card command
PKCS	Public-Key Cryptography Standards
PIN	Personal Identification Number
PIV	Personal Identity Verification
PIX	Proprietary Identifier extension
PUK	PIN Unblocking Key
RFU	Reserved for Future Use
RID	Registered application provider Identifier
RSA	Rivest, Shamir, Adleman
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, …, 9, A, B, C, D, E, and F. A byte consists of two hexadecimal digits, for example, '2D'. A sequence of bytes may be enclosed in single quotation marks, for example 'A0 00 00 01 16' rather than given as a sequence of individual bytes, 'A0' '00' '00' '01' '16'.

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

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

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

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

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

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

Appendix C—References

[1] Federal Information Processing Standard 201-1, Change Notice 1, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, March 2006. (See http://csrc.nist.gov)

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

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

[4] 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).*

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

[6] NIST Special Publication 800-78-2, Cryptographic *Algorithms and Key Sizes for Personal Identity Verification*, February 2010. (See http://csrc.nist.gov)

[7] NIST Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, March 2007. (See http://csrc.nist.gov)

[8] Standards for Efficient Cryptography Group (SECG), “SEC 1: Elliptic Curve Cryptography”, Version 1.0, September 2000.

[9] Jakob Jonsson and Burt Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003. (See http://tools.ietf.org/html/rfc3447)

NIST Special Publication 800-73-3

Interfaces for Personal Identity Verification – Part 3: End-Point PIV Client Application Programming Interface

Ramaswamy Chandramouli
David Cooper
James F. Dray
Hildegard Ferraiolo
Scott B. Guthery
William MacGregor
Ketan Metha

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

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

February 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

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

National Institute of Standards and Technology Special Publication 800-73-3, Part 3, 20 pages (February 2010)

Acknowledgements

The authors (Ramaswamy Chandramouli, David Cooper, James Dray, Hildegard Ferraiolo, William MacGregor, of NIST, Ketan Mehta of Booz Allen Hamilton and Scott Guthery of HID Global) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Government Smart Card Interagency Advisory Board (GSC-IAB) and InterNational Committee for Information Technology Standards (INCITS) for providing detailed technical inputs to the SP 800-73 development process. The authors also gratefully acknowledge and appreciate the many contributions from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

Table of Contents

List of Appendices

List of Tables

1. Introduction

The Homeland Security Presidential Directive 12 (HSPD-12) called for a common identification standard to be adopted governing the interoperable use of identity credentials to allow physical and logical access to Federal government locations and systems. The Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201 (FIPS 201) [1] was developed to establish standards for identity credentials. Special Publication 800-73-3 (SP 800-73-3) specifies interface requirements for retrieving and using the identity credentials from the PIV Card and is a companion document to FIPS 201.

1.1 Authority

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

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

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

1.2 Purpose

FIPS 201 defines procedures for the PIV lifecycle activities including identity proofing, registration, PIV Card issuance, and PIV Card usage. FIPS 201 also specifies that the identity credentials must be stored on a smart card. SP 800-73-3 contains technical specifications to interface with the smart 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 (API). Moreover, SP 800-73-3 enumerates requirements where the standards include options and branches. The specifications go further by constraining implementers' interpretations of the normative standards. Such restrictions are designed to ease implementation, facilitate interoperability, and ensure performance in a manner tailored for PIV applications.

1.3 Scope

SP 800-73-3 specifies the PIV data model, Application Programming Interface and card interface requirements necessary to comply with the use cases, as defined in Section 6 of FIPS 201 and further elaborated in Appendix B of SP 800-73-3, Part 1. Interoperability is defined as the use of PIV identity credentials such that client application programs, compliant card applications, and compliant integrated circuits cards (ICC) can be used interchangeably by all information processing systems across Federal agencies.

This Part, Special Publication 800-73-3 (SP 800-73-3) Part 3: *End-Point PIV Client Application Programming Interface* contains technical specifications of the PIV client application programming interface to the PIV Card.

1.4 Audience and Assumptions

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

Readers should also be aware of SP 800-73-3 Part 1, Section I, which details the Revision History of SP800-73-3, Section II which contains Configuration Management Recommendations and Section III which specifies NPIVP Conformance Testing Procedures.

1.5 Content and Organization

All sections in this document are *normative* (i.e., mandatory for compliance) unless specified as *informative* (i.e., non-mandatory). Following is the structure of Part 3:

+ Section 1, *Introduction*, provides the purpose, scope, audience and assumptions of the document and outlines its structure.

+ Section 2, *Overview: End-Point Concepts and Constructs*, describes both the PIV Card Application and the PIV client application programming interface. This section is informative.

+ Section 3, *End-Point 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, *Terms, Acronyms, and Notation*, contains the list of Terms and Acronyms used in this document and explains the notation in use. This section is informative.

+ Appendix B, *References*, contains the list of documents used as references by this document. This section is informative.

2. Overview: End-Point Concepts and Constructs

SP 800-73-3 Part 2 and Part 3 define two interfaces to an ICC that contains the Personal Identity Verification card application: a low-level PIV Card Application card command interface (Part 2) and a high-level PIV client-API (Part 3).

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

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

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

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

3. End-Point Client Application Programming Interface

Table 1 lists the entry points on the PIV client application programming interface. This section references Object Identifiers (OIDs), which are defined and can be found in Part 1 (Table 2).

Table 1. Entry Points on PIV End-Point Client Application Programming Interface

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

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 800-73-3 Part 3 conformant PIV Middleware, the parameter returns "800-73-3 Client API".

+ For SP 800-73-2 Part 3 conformant PIV Middleware, the parameter returns **"**800-73-2 Client API"**.**

+ For SP 800-73-1 conformant PIV Middleware**,** the pivMiddlewareVersion Client API function is not supported. Therefore, a client application invoking the pivMiddlewareVersion function should expect a "function-not-supported" error from a SP 800-73-1 conformant PIV Middleware. For purposes of version determination, failure to obtain a specific version from pivMiddlewareVersion shall be considered equivalent to obtaining a response of "800-73-1 Client API".

Return Codes: `PIV_OK`

SP 800-73-3 Part 3 conformant PIV Middleware shall implement all PIV Middleware functions listed in Table 1 and be able to recognize and process all mandatory and optional PIV data objects.

Note: Only SP 800-73-3 based PIV Middleware can recognize, store, and retrieve new optional data objects and/or features that have been introduced for PIV Cards in Part 1 of SP 800-73-3. SP 800-73-1 or SP 800-73-2 based PIV Middleware remain valid implementations; however, Agencies are cautioned that using these implementations may result in limited interoperability. Further information can be found in Part 1 of SP 800-73-3. It provides a SP 800-73 Revision History (Section I) and recommendations for PIV Middleware Configuration Management (Section II).

3.1.2 pivConnect

Purpose: Connects the client application programming interface 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 establish concurrent connections to the ICC. If FALSE and the connection is established then the calling client application has exclusive access to the ICC.

`connectionDescription` A connection description data object (tag '7F 21'). See Table 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.

The connection description BER-TLV [2] used on the PIV client application programming interface shall have the structure described in Table 2.

Table 2. Data Objects in a Connection Description Template (Tag '7F21')

Description	Tag	M/O/C[1]	Comment
Interface device – PC/SC	'81'	C	Card reader name
Interface device – SCP	'82'	C	Card reader identifier on terminal equipment
Interface device – EMR	'83'	C	Contactless connection using radio transmission
Interface device – IR	'84'	C	Contactless connection using infrared transmission
Interface device – PKCS#11	'85'	C	PKCS#11 interface
Interface device – CryptoAPI	'86'	C	CryptoAPI interface
Network node – Local	'90'	C	No network between client application host and card reader host
Network node – IP	'91'	C	IP address of card reader host
Network node – DNS	'92'	C	Internet domain name of card reader host
Network node – ISDN	'93'	C	ISDN dialing number string of terminal equipment containing the card reader

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

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

When used as an argument to the pivConnect entry point on the PIV client application programming interface described in this section, an '8x' series data object with zero length together with a '9x' series data object requests 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 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
```

[1] M = Mandatory, O = Optional, C = Conditional. For the definition of M/O/C see Appendix A.3.

3.1.3 pivDisconnect

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

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

Parameters:

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

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

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 Part 2, Table 3.
APLength	Length of the application properties.

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

3.2.2 pivLogIntoCardApplication

Purpose: Set 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 security state/status in the PIV Card Application context.

The authenticator BER-TLV used on the PIV client application programming interface 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	E.g. the PIN value or challenge response
Key reference	'83'	M	See Part 1, Table 3 for PIN Key Reference values

Return Codes:
```
PIV_OK
PIV_INVALID_CARD_HANDLE
PIV_AUTHENTICATOR_MALFORMED
PIV_AUTHENTICATION_FAILURE
PIV_CARD_READER_ERROR
```

3.2.3 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; for example, '2.16.840.1.101.3.7.1.1.2.2.1'. See Part 1, Table 2.
oidLength	Length of the object identifier.
data	Retrieved data content.
DataLength	Length of the data to retrieve 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
```

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[2] such as encryption or signing on a sequence of bytes. Part 1, 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,
```

[2] 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 on the client side are performed outside the PIV Middleware.

```
   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. See Tables 6-2 and 6-3 in SP 800-78 [3].
keyReference	Identifier of the on-card key to be used for the cryptographic operation. + See Tables 6-1 and 6-3 in SP 800-78 for key reference values. + See Part 1, Table 6 for key reference values of retired private Key Management Keys.
algorithmInput	Sequence of bytes used as the input to the cryptographic operation.[3]
inputLength	Length of the algorithm input.
algorithmOutput	Sequence of bytes output by the cryptographic operation.
outputLength	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
```

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.

3.4 Entry Points for Credential Initialization and Administration

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
```

[3] The algorithmInput for RSA algorithms shall be restricted to the range 0 to *n*-1, where *n* is the RSA modulus.

```
);
```

Parameters:

cardHandle	Opaque identifier of the card to be acted upon as returned by pivConnect.
OID	Object identifier of the object whose data content is to be replaced coded as a string; for example, "2.16.840.1.101.3.7.1.1.2.2.1". See Part 1, Table 2.
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
```

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 Part 1, Table 4.
publicKey	BER-TLV data objects defining the public key of the generated key pair. See Part 2, Table 10.
KeyLength	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_INVALID_KEY_OR_KEYALG_COMBINATION
PIV_UNSUPPORTED_CRYPTOGRAPHIC_MECHANISM
PIV_CARD_READER_ERROR
```

Appendix A—Terms, Acronyms, and Notation

A.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.
Algorithm Identifier	A PIV algorithm identifier is a one-byte identifier that specifies a cryptographic algorithm and key size. For symmetric cryptographic operations, the algorithm identifier also specifies a mode of operation (ECB).
BER-TLV Data Object	A data object coded according to ISO/IEC 8825-2.
Card	An integrated circuit card.
Card Application	A set of data objects and card commands that can be selected using an application identifier.
Card Interface Device	An electronic device that connects an integrated circuit card and the card applications therein to a client application.
Card Reader	Synonym for card interface device.
Client Application	A computer program running on a computer in communication with a card interface device.
Data Object	An item of information seen at the card command interface for which are specified a name, a description of logical content, a format and a coding.
Interface Device	Synonym for card interface device.
Key Reference	A PIV key reference is a one-byte identifier that specifies a cryptographic key according to its PIV Key Type. The identifier used in cryptographic protocols 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 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.

A.2 Acronyms

AID	Application Identifier
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
FIPS	Federal Information Processing Standards
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
OID	Object Identifier
OMB	Office of Management and Budget
PC/SC	Personal Computer/Smart Card
PIN	Personal Identification Number
PIV	Personal Identity Verification

PIX	Proprietary Identifier eXtension
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
RFU	Reserved for Future Use
RID	Registered application provider IDentifier
SP	Special Publication
TLV	Tag-Length-Value

A.3 Notation

The sixteen 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'. A sequence of bytes may be enclosed in single quotation marks, for example 'A0 00 00 01 16' rather than given as a sequence of individual bytes, 'A0' '00' '00' '01' '16'.

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

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

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

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

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

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

Appendix B—References

[1] Federal Information Processing Standard 201-1, Change Notice 1, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, March 2006. (See http://csrc.nist.gov)

[2] 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).*

[3] NIST Special Publication 800-78-2, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification*, February 2010. (See http://csrc.nist.gov)

NIST Special Publication 800-73-3

Interfaces for Personal Identity Verification – Part 4: The PIV Transitional Interfaces and Data Model Specification

Ramaswamy Chandramouli
David Cooper
James F. Dray
Hildegard Ferraiolo
Scott B. Guthery
William MacGregor
Ketan Mehta

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

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

February 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

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

National Institute of Standards and Technology Special Publication 800-73-3, Part 4, 20 pages, (February 2010)

Acknowledgements

The authors (Ramaswamy Chandramouli, David Cooper, James Dray, Hildegard Ferraiolo, William MacGregor of NIST, Ketan Mehta of Booz Allen Hamilton, and Scott Guthery of HID Global) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. Special thanks to the Government Smart Card Interagency Advisory Board (GSC-IAB) and InterNational Committee for Information Technology Standards (INCITS) for providing detailed technical inputs to the SP 800-73 development process. The authors also gratefully acknowledge and appreciate the many contributions from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

TABLE OF CONTENTS

List of Appendices

List of Tables

1. Introduction

The Homeland Security Presidential Directive 12 (HSPD-12) called for a common identification standard to be adopted governing the interoperable use of identity credentials to allow physical and logical access to Federal government locations and systems. The Personal Identity Verification (PIV) of Federal Employees and Contractors, Federal Information Processing Standard 201 (FIPS 201) [1] was developed to establish standards for identity credentials. Special Publication 800-73-3 (SP 800-73-3) specifies interface requirements for retrieving and using the identity credentials from the PIV Card[1] and is a companion document to FIPS 201.

1.1 Authority

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

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

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

1.2 Purpose

FIPS 201 defines procedures for the PIV lifecycle activities including identity proofing, registration, PIV Card issuance, and PIV Card usage. FIPS 201 also specifies that the identity credentials must be stored on a smart card. SP 800-73-3 contains technical specifications to interface with the smart 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, SP 800-73-3 enumerates requirements where the standards include options and branches. The specifications go further by constraining implementers' interpretations of the normative standards. Such restrictions are designed to ease implementation, facilitate interoperability, and ensure performance in a manner tailored for PIV applications.

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

1.3 Scope

SP 800-73-3 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 elaborated in Appendix B of SP 800-73-3 Part 1. Interoperability is defined as the use of PIV identity credentials such that client-application programs, compliant card applications, and compliant integrated circuits cards (ICC) can be used interchangeably by all information processing systems across Federal agencies.

Parts 2, 3, and 4 of SP 800-73-3 describe two realizations of the client application programming and card command interfaces for Personal Identity Verification: the transitional interfaces (this Part 4) and the end-point interfaces (Parts 2 and 3). The transitional interface may be used by agencies with an existing identity card program as an optional intermediate step in evolving to the end-point interfaces.

This part, Special Publication 800-73-3, Part 4*: The PIV Transitional Interfaces and Data Model Specification*, contains informative links to specifications of the transitional PIV Card command interface and client application programming interface of the transitional PIV Card. Part 4 also describes the PIV Data Model that is common between End-Point and transitional interface specifications.

1.4 Audience and Assumptions

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

Readers should also be aware of and familiar with the Revision History section of SP 800-73-3 Part 1.

1.5 Document Overview and Structure

All sections in this document are *normative* (i.e., mandatory for compliance) unless specified as *informative* (i.e., non-mandatory). Following is the structure of this document:

+ Section 1: *Introduction,* provides the purpose, scope, audience, and assumptions of the document and outlines its structure.
+ Section 2: *Overview and Migration Considerations,* provides the specification that is common to both the transitional and end-point interfaces. Section 2 also includes guidelines as to strategies for migrating from the transitional interfaces to the end-point interfaces.
+ Section 3: *Transition Card Interfaces,* provides links to transitional interface specifications that are implemented today by agencies with legacy GSC-IS based card deployments. This section is informative.
+ Appendix A: *Terms, Acronyms, and Notation,* contains the list of Terms and Acronyms used in this document and explains notation in use. This section is informative.
+ Appendix B: *References,* contains the list of documents used as references by this document. This section is informative.

2. Overview and Migration Considerations

2.1 Migration Considerations

SP 800-73-3 Parts 1 – 4 provide two interface specifications: 1) a transitional card specification as described in this Part 4; and 2) a FIPS 201 End-Point card specification as described in Parts 1 – 3 of SP 800-73-3. Part 4 interface specifications are informative PIV profiles derived from the Government Smart Card Interoperability Specification (GSC-IS), Version 2.1 [2]. It presents one possible path that agencies with existing GSC-IS based smart card deployments may choose to follow during the transition to End-Point PIV Card deployment. All agencies must ultimately comply with End-Point specifications in accordance with the schedule provided by the Office of Management and Budget (OMB). End-Point deployment is therefore the end state of each agency's transition plan.

Agencies may either elect to implement an approved transitional specification as specified in this document (Part 4), particularly when migrating from currently widely implemented identity card architectures to the End-Point specifications described in Parts 1 – 3 of SP 800-73-3, or to implement the End-Point specification directly. NIST supports agency efforts towards government-wide PIV-End-Point interoperability described in the Parts 1 – 3 specification. NIST also supports transition specifications of Part 4 for widely implemented deployments as they migrate towards the End-Point specifications.

The migration path to End-Point implementation is based on continuity of the PIV data model. Exactly the same data appear on both the transitional and end-point interfaces. Therefore, description of the data for personal identity verification, the PIV data model, is duplicated from Part 1 (Section 3) in Section 2.2 below[2].

Specific considerations associated with this migration path are highlighted below:

+ The transitional specifications present a subset of the dual GSC-IS card edge interfaces. The End-Point specifications present a unified card edge interface that is technology independent and compliant with existing international standards.

+ The End-Point specifications provide limited credential administration functionality. A unified and interoperable card management solution between issuing domains including the loading of new card applications is not provided.

+ Named data objects within the data model may be directly accessed. If a data object is managed by the default application, it can be retrieved directly without selecting the application. This avoids a requirement to search through discovery to get named data objects. Otherwise, the (non-default) application managing the data object is selected and the data object is retrieved from this application. The GET DATA command described in Part 2 retrieves a data object without prior selection of the file containing the content of the data object.

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

[2] Although the same data objects are present on the end-point and transitional interfaces, different representations for the same data objects may be used.

will be explicitly managed by NIST and data object identifiers of GSC-IS and transitional systems in the range 'A000' through 'FFFF' are placed under control of the card issuer.

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

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

2.2 PIV Data Model

Table 1 defines a high level view of the data model. Each on-card storage container is labeled either as Mandatory (M) or Optional (O). This data model is designed to enable and support dual interface cards. Note that access conditions based on the interface mode (contact vs. contactless) take precedence over all Access Rules defined in Table 1, Column 3.

Table 1. Data Model Containers

Container Name	Container ID	Access Rule for Read	Contact / Contactless[3]	M/O
Card Capability Container	0xDB00	Always	Contact	M
Cardholder Unique Identifier	0x3000	Always	Contact and Contactless	M
X.509 Certificate for PIV Authentication	0x0101	Always	Contact	M
Cardholder Fingerprints	0x6010	PIN	Contact	M
Security Object	0x9000	Always	Contact	M
Cardholder Facial Image	0x6030	PIN	Contact	O
Printed Information	0x3001	PIN	Contact	O
X.509 Certificate for Digital Signature	0x0100	Always	Contact	O
X.509 Certificate for Key Management	0x0102	Always	Contact	O
X.509 Certificate for Card Authentication	0x0500	Always	Contact and Contactless	O
Discovery Object	0x6050	Always	Contact and Contactless	O
Key History Object	0x6060	Always	Contact	O
Retired X.509 Certificate for Key Management 1	0x1001	Always	Contact	O
Retired X.509 Certificate for Key Management 2	0x1002	Always	Contact	O
Retired X.509 Certificate for Key Management 3	0x1003	Always	Contact	O
Retired X.509 Certificate for Key Management 4	0x1004	Always	Contact	O
Retired X.509 Certificate for Key Management 5	0x1005	Always	Contact	O
Retired X.509 Certificate for Key Management 6	0x1006	Always	Contact	O

[3] Contact interface mode means the container is accessible through contact interface only. Contact and contactless interface mode means the container can be accessed from either interface.

Container Name	Container ID	Access Rule for Read	Contact / Contactless[3]	M/O
Retired X.509 Certificate for Key Management 7	0x1007	Always	Contact	O
Retired X.509 Certificate for Key Management 8	0x1008	Always	Contact	O
Retired X.509 Certificate for Key Management 9	0x1009	Always	Contact	O
Retired X.509 Certificate for Key Management 10	0x100A	Always	Contact	O
Retired X.509 Certificate for Key Management 11	0x100B	Always	Contact	O
Retired X.509 Certificate for Key Management 12	0x100C	Always	Contact	O
Retired X.509 Certificate for Key Management 13	0x100D	Always	Contact	O
Retired X.509 Certificate for Key Management 14	0x100E	Always	Contact	O
Retired X.509 Certificate for Key Management 15	0x100F	Always	Contact	O
Retired X.509 Certificate for Key Management 16	0x1010	Always	Contact	O
Retired X.509 Certificate for Key Management 17	0x1011	Always	Contact	O
Retired X.509 Certificate for Key Management 18	0x1012	Always	Contact	O
Retired X.509 Certificate for Key Management 19	0x1013	Always	Contact	O
Retired X.509 Certificate for Key Management 20	0x1014	Always	Contact	O
Cardholder Iris Images	0x1015	PIN	Contact	O

Part 1, Appendix A provides a detailed spreadsheet for the data model. ContainerIDs and Tags within the containers for each data object are defined by this data model and in accordance with SP 800-73-3 naming conventions.

A PIV Card Application shall contain five mandatory interoperable data objects and may contain twenty-eight optional interoperable data objects. The five mandatory data objects for interoperable use are as follows:

1. Card Capability Container
2. Card Holder Unique Identifier
3. X.509 Certificate for PIV Authentication
4. Cardholder Fingerprints
5. Security Object

The twenty-eight optional data objects for interoperable use are as follows:

1. Cardholder Facial Image
2. Printed Information
3. X.509 Certificate for Digital Signature
4. X.509 Certificate for Key Management
5. X.509 Certificate for Card Authentication
6. Discovery Object
7. Key History Object
8. 20 retired X.509 Certificates for Key Management
9. Cardholder Iris Images

2.3 Mandatory Data Elements

The five mandatory data objects support FIPS 201 minimum mandatory compliance.

2.3.1 Card Capability Container

The Card Capability Container (CCC) is a mandatory data object whose purpose is to facilitate compatibility of GSC-IS applications with End-Point PIV Cards.

The CCC supports minimum capability for retrieval of the data model and optionally the application information as specified in Government Smart Card Interoperability Specification (GSC-IS) [2]. The data model of the PIV Card Application shall be identified by data model number 0x10. Deployed applications use 0x00 through 0x04. This enables the GSC-IS application domain to correctly identify a new data model namespace and structure as defined in this document.

For End-Point PIV Card Applications, the PIV data objects exist in a namespace tightly managed by NIST and a CCC discovery mechanism is not needed by End-Point applications. Therefore, all data elements of the CCC, except for the data model number, may optionally have a length value set to zero bytes (i.e., no value field will be supplied). The content of the CCC data elements, other than the data model number, are out of scope for this specification.

2.3.2 Card Holder Unique Identifier

The Card Holder Unique Identifier (CHUID) data object is defined in accordance with the Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems (TIG SCEPACS) [4]. For this specification, the CHUID is common between the contact and contactless chips. For dual chip implementations, the CHUID is copied in its entirety between the two chips.

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

+ The Buffer Length field is an optional TLV element. This element is the length in bytes of the entire CHUID, excluding the Buffer Length element itself, but including the CHUID's Asymmetric Signature element. The calculation of the asymmetric signature must exclude the Buffer Length element if it is present.

+ The Federal Agency Smart Credential Number (FASC-N) shall be in accordance with TIG SCEPACS [4]. A subset of the FASC-N, the FASC-N Identifier, shall be the unique identifier as described in [4, 6.6]: "The combination of an Agency Code, System Code, and Credential Number is a fully qualified number that is uniquely assigned to a single individual". The Agency Code is assigned to each Department or Agency by Special Publication 800-87 (SP 800-87), *Codes for Identification of Federal and Federally-Assisted Organizations* [5]. The subordinate System Code and Credential Number value assignment is subject to Department or Agency policy, provided that the FASC-N identifier (i.e., the concatenated Agency Code, System Code, and Credential Number) is unique for each card. The same FASC-N value shall be used in all the PIV data objects that include the FASC-N. To eliminate unnecessary use of the SSN[4], the FASC-N's Person Identifier (PI) field should not encode the SSN. TIG SCEPACS also specifies PACS interoperability requirements in Section 2.1, 10th paragraph of [4, 2.1]: "For full interoperability of a PACS it must at a minimum be able to distinguish fourteen digits (i.e., a combination of an Agency Code, System Code, and Credential Number) when matching FASC-N based credentials to enrolled card holders."

[4] See the attachment to OMB M-07-16, Section 2: "Reduce the Use of Social Security Numbers".

+ The Global Unique Identification number (GUID) field must be present, and may include either a UUID (see Section 3.3), an issuer assigned IPv6 address[5], or be coded as all zeros (0x00).

+ The DUNS and Organizational Code fields are optional.

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

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

2.3.3 X.509 Certificate for PIV Authentication

The X.509 Certificate for PIV Authentication and its associated private key, as defined in FIPS 201, is used to authenticate the card and the cardholder. The read access control rule for the X.509 Certificate for PIV Authentication is "Always," meaning the certificate can be read without access control restrictions. The Public Key Infrastructure (PKI) cryptographic function (see Table 3, Part 1) is protected with a "PIN" access rule. In other words, private key operations using the PIV Authentication Key require the Personal Identification Number (PIN) to be submitted, but a successful PIN submission enables multiple private key operations without additional cardholder consent.

2.3.4 Cardholder Fingerprints

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

2.3.5 Security Object

The Security Object is in accordance with Appendix C of PKI for Machine Readable Travel Documents (MRTD) Offering ICC Read-Only Access Version 1.1 [6]. Tag 0xBA is used to map the ContainerIDs in the PIV data model to the 16 Data Groups specified in the MRTD. The mapping enables the Security Object to be fully compliant for future activities with identity documents.

The "DG-number-to-Container-ID" mapping object TLV in tag 0xBA encapsulates a series of three byte triples - one for each PIV data object included in the Security Object. The first byte is the Data Group (DG) number, and the second and third bytes are the most and least significant bytes (respectively) of the Container ID value. The DG number assignment is arbitrary; however, the same number assignment applies to the DataGroupNumber(s) in the DataGroupHash(es). This will ensure that the ContainerIDs in the mapping object refer to the correct hash values in the Security Object (0xBB).

The 0xBB Security Object is formatted according to the MRTD [6, Appendix C]. The LDS Security Object itself must be in ASN.1 DER format, formatted as specified in [6, Appendix C.2]. This

[5] The use of IPv6 addresses in the GUID field is deprecated. It will be removed in a future revision of SP 800-73.

structure is then inserted into the encapContentInfo field of the Cryptographic Message Syntax (CMS) object specified in [6, Appendix C.1].

The card issuer's digital signature key used to sign the CHUID shall also be used to sign the Security Object. The signature field of the Security Object, tag 0xBB, shall omit the issuer's certificate, since it is included in the CHUID. At a minimum, unsigned data objects, such as the Printed Information data object, shall be included in the Security Object if present. For maximum protection against credential splicing attacks (credential substitution), it is recommended, however, that all PIV data objects, except the PIV X.509 certificates, be included in the Security Object.

2.4 Optional Data Elements

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

2.4.1 Cardholder Facial Image

The photo on the chip supports human verification only. It is not intended to support facial recognition systems for automated identity verification.

2.4.2 Printed Information

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

2.4.3 X.509 Certificate for Digital Signature

The X.509 Certificate for Digital Signature and its associated private key, as defined in FIPS 201, support the use of digital signatures for the purpose of document signing. The read access control rule for the X.509 Certificate is "Always", meaning the certificate can be read without access control restrictions. The PKI cryptographic function is protected with a "PIN Always" access rule. In other words, the PIN must be submitted every time immediately before a *Digital Signature Key* operation. This ensures cardholder participation every time the private key is used for digital signature generation.

2.4.4 X.509 Certificate for Key Management

The X.509 Certificate for Key Management and its associated private key, as defined in FIPS 201, support the use of encryption for the purpose of confidentiality. This key pair may be escrowed by the issuer for key recovery purposes. The read access control rule for the X.509 Certificate is "Always", meaning the certificate can be read without access control restrictions. The PKI cryptographic function is protected with a "PIN" access rule. In other words, once the PIN is submitted, subsequent *Key Management Key* operations can be performed without requiring the PIN again. This enables multiple private key operations without additional cardholder consent.

2.4.5 X.509 Certificate for Card Authentication

FIPS 201 specifies the optional Card Authentication Key (CAK) as an asymmetric or symmetric key that is used to support additional physical access applications. For an asymmetric CAK, the read

access control rule of the corresponding X.509 Certificate for Card Authentication is "Always", meaning the certificate can be read without access control restrictions. Private (asymmetric) key operations or secret (symmetric) key operations are defined as "Always". In other words, the private or secret key can be used without access control restrictions. If the CAK is implemented, an asymmetric or symmetric CAK is generated by the PIV Card Issuer in accordance with FIPS 140-2 requirements for key generation. A CAK may be generated on-card or off-card. If a CAK is generated off-card, the result of each key generation will be injected into at most one PIV Card.

2.4.6 Discovery Object

The Discovery Object, if implemented, is the 0x7E interindustry ISO/IEC 7816-6 template that nests interindustry data objects. For the Discovery Object, the 0x7E template nests two BER-TLV structured interindustry data elements: 1) tag 0x4F contains the AID of the PIV Card Application and 2) tag 0x5F2F lists the PIN Usage Policy.

+ Tag 0x4F encodes the PIV Card Application AID as follows:

  {'4F 0B A0 00 00 03 08 00 00 10 00 01 00'}

+ Tag 0x5F2F encodes the PIN Usage Policy as follows:

  First byte: 0x40 indicates that the PIV Card Application PIN alone satisfies the PIV Access Control Rules (ACRs) for command execution[6] and object access.

  0x60 indicates that both the PIV Card Application PIN and Global PIN satisfy the PIV ACRs for command execution and PIV data object access.

  Bits 5 through 1 of the first byte are RFU.

  The second byte of the PIN Usage Policy encodes the cardholder's PIN preference for PIV Cards with both the PIV Card Application PIN and the Global PIN enabled:

  Second byte: 0x10 indicates that the PIV Card Application PIN is the primary PIN used to satisfy the PIV ACRs for command execution and object access.

  0x20 indicates that the Global PIN is the primary PIN used to satisfy the PIV ACRs for command execution and object access.

Note: If the first byte is set to 0x40, then the second byte is RFU and shall be set to 0x00.

PIV Card Applications that satisfy the PIV ACRs for PIV data object access and command execution[7] with both the PIV Card Application PIN and Global PIN shall implement the Discovery Object with the PIN Usage Policy set to 0x60 zz where zz is set to either 0x10 or 0x20.

The encoding of the 0x7E Discovery Object is as follows:

{'7E 12' {{'4F 0B A0 00 00 03 08 00 00 10 00 01 00'} {'5F 2F 02 xx yy'}}}, where xx and yy encode the first and second byte of the PIN Usage Policy as described in this section.

[6] Command execution pertains to the VERIFY APDU and optionally to the CHANGE REFERENCE DATA APDU.
[7] Command execution pertains to the VERIFY APDU and optionally to the CHANGE REFERENCE DATA APDU.

The Security Object enforces integrity of the Discovery Object according to the issuer.

2.4.7 Key History Object

Up to twenty retired Key Management private keys may be stored in the PIV Card Application. The Key History object provides information about the retired Key Management private keys that are present within the PIV Card Application. Retired Key Management private keys are private keys that correspond to X.509 certificates for Key Management that have expired, have been revoked, or have otherwise been superseded. The Key History object shall be present in the PIV Card Application if the PIV Card Application contains any retired Key Management private keys, but may be present even if no such keys are present in the PIV Card Application. For each retired Key Management private key in the PIV Card Application, the corresponding certificate may either be present within the PIV Card Application or may only be available from an on-line repository.

The Key History object includes two mandatory fields, *keysWithOnCardCerts* and *keysWithOffCardCerts*, and one optional field, *offCardCertURL*. The *keysWithOnCardCerts* field indicates the number of retired private keys within the PIV Card Application for which the corresponding certificates are also stored within the PIV Card Application. The *keysWithOffCardCerts* field indicates the number of retired private keys within the PIV Card Application for which the corresponding certificates are not stored within the PIV Card Application. The numeric values in both *keysWithOnCardCerts* and *keysWithOffCardCerts* are represented as unsigned binary integers. The *offCardCertURL* field contains a URL that points to a file containing the certificates corresponding to all of the retired private keys within the PIV Card Application, including those for which the corresponding certificate is also stored within the PIV Card Application. The *offCardCertURL* field shall be present if the *keysWithOffCardCerts* value is greater than zero and shall be absent if the values of both *keysWithOnCardCerts* and *keysWithOffCardCerts* are zero. The *offCardCertURL* field may be present if the *keysWithOffCardCerts* value is zero but the *keysWithOnCardCerts* value is greater than zero.

The file that is pointed to by the *offCardCertURL* field shall contain the DER encoding of the following data structure:

```
OffCardKeyHistoryFile ::= SEQUENCE SIZE (1..20) OF SEQUENCE {
        keyReference        OCTET STRING (SIZE(1))
        cert                Certificate
}
```

where **keyReference** is the key reference for the private key on the card and **cert** is the corresponding X.509 certificate.[8] The *offCardCertURL* field shall have the following format:

"http://" <DNS name> "/" <ASCII-HEX encoded SHA-256 hash [13] of **OffCardKeyHistoryFile**>

The private keys for which the corresponding certificates are stored within the PIV Card Application shall be assigned to the lowest numbered key references reserved for retired Key Management private keys. For example if *keysWithOnCardCerts* is 5, then the corresponding private keys shall be assigned to key references '82', '83', '84', '85', and '86'.

The private keys for which the corresponding certificates are not stored within the PIV Card Application shall be assigned to the highest numbered key references reserved for retired Key

[8] The ASN.1 for **Certificate** may be imported from the ASN.1 module **PKIX1Explicit88** in Appendix A.1 of RFC 5280 [12].

Management private keys. For example, if *keysWithOffCardCerts* is 3, then the corresponding private keys shall be assigned to key references '93', '94', and '95'.

Private keys do not have to be stored within the PIV Card Application in the order of their age. However, if the certificates corresponding to only some of the retired Key Management private keys are available within the PIV Card Application then the certificates that are stored in the PIV Card Application shall be the ones that were most recently issued.

The Key History object is only available over the contact interface. The read access control rule for the Key History object is "Always", meaning that it can be read without access control restrictions.

The Security Object enforces integrity of the Key History object according to the issuer.

2.4.8 Retired X.509 Certificates for Key Management

These objects hold the X.509 certificates for Key Management corresponding to retired Key Management Keys, as described in Section 2.4.7. Retired Key Management private keys and their corresponding certificates are only available over the contact interface. The read access control rule for these certificates is "Always", meaning the certificates can be read without access control restrictions. The PKI cryptographic function for all of the retired Key Management Keys is protected with a "PIN" access rule. In other words, once the PIN is submitted and verified, subsequent *Key Management Key* operations can be performed with any of the retired Key Management Keys without requiring the PIN again. This enables multiple private key operations without additional cardholder consent.

2.4.9 Cardholder Iris Images

The iris data object specifies compact images of the cardholder's irises. The images are suitable for use in iris recognition systems for automated identity verification.

2.5 Inclusion of Universally Unique IDentifiers (UUIDs)

As defined in [9], the presence of a Universally Unique IDentifier (UUID) conformant to the specification [10] is required in each identification card issued by Non-Federal Issuers, referred to as "PIV Interoperable" (PIV-I) or "PIV Compatible" (PIV-C) cards. The intent of [9] is to enable issuers to issue cards that are technically interoperable with Federal PIV Card readers and applications, and that may be trusted for particular purposes through a decision of the relying Federal Department or Agency. Because the goal is interoperability of PIV-I and PIV-C cards with the Federal PIV System, the technical requirements for the inclusion of the UUID are specified in this document. To include a UUID identifier on a PIV-I, PIV-C, or PIV Card, a credential issuer shall meet the following specifications for all relevant data objects present on an issued identification card.

1. If the card is a PIV-I or PIV-C card, the FASC-N in the CHUID shall have Agency Code equal to 9999, System Code equal to 9999, and Credential Number equal to 999999, indicating that a UUID is the primary credential identifier. In this case, the FASC-N shall be omitted from certificates and CMS-signed data objects. If the card is a PIV Card, the FASC-N in the CHUID shall be populated as described in Section 2.3.2, and the FASC-N shall be included in authentication certificates and CMS-signed data objects as required by FIPS 201.

2. The value of the GUID data element of the CHUID data object shall be a 16-byte binary representation of a valid UUID [10]. The UUID should be version 1, 4, or 5, as specified in [10], Section 4.1.3.

3. The same 16-byte binary representation of the UUID value shall be present as the value of an entryUUID attribute, as defined in [11], in any CMS-signed data object that is required to contain a pivFASC-N attribute on a PIV Card, i.e., in the fingerprint template and facial image data objects, if present.

4. The string representation of the same UUID value shall be present in the PIV Authentication Certificate and the Card Authentication Certificate, if present, in the subjectAltName extension encoded as a URI, as specified by [10], Section 3.

The option specified in this section supports the use of UUIDs by Non-Federal Issuers. It also allows, but does not require, the use of UUIDs as optional data elements on PIV Cards. PIV Cards must meet all requirements in FIPS 201 whether or not the UUID identifier option is used; in particular, the FASC-N identifier must be present in all PIV data objects as specified by FIPS 201 and its normative references. PIV Cards that include UUIDs must include the UUIDs in all data objects described in (2) through (4).

3. Transition Card Interfaces

3.1 Middleware Application Programming Interface

Reference [7] is an example of a transitional (GSC-IS) middleware API specification.

3.2 Card Edge Commands

Reference [8] is an example of a transitional (GSC-IS) card edge command specification.

Appendix A—Terms, Acronyms, and Notation

A.1 Terms

Card	An integrated circuit card.
Card Application	A set of data objects and card commands that can be selected using an application identifier.
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.

A.2 Acronyms

ACR	Access Control Rule
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
BSI	Basic Services Interface
CAK	Card Authentication Key
CBEFF	Common Biometric Exchange Formats Framework
CCC	Card Capability Container
CHUID	Card Holder Unique Identifier
CMS	Cryptographic Message Syntax
DG	Data Group
FASC-N	Federal Agency Smart Credential Number
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GSC-IAB	Government Smart Card Interagency Advisory Board
GSC-IS	Government Smart Card Interoperability Specification

GUID	Global Unique Identification Number
HSPD	Homeland Security Presidential Directive
ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
INCITS	InterNational Committee for Information Technology Standards
ISO	International Organization for Standardization
ITL	Information Technology Laboratory
LSB	Least Significant Bit
MRTD	Machine Readable Travel Document
MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
PACS	Physical Access Control System
PI	Person Identifier, a field in the FASC-N
PIN	Personal Identification Number
PIV	Personal Identity Verification
PIV-C	PIV Compatible
PIV-I	PIV Interoperable
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
RFU	Reserved for Future Use
RID	Registered application provider IDentifier
RSA	Rivest, Shamir, Adleman
SCEPACS	Smart Card Enabled Physical Access Control System
SHA	Secure Hash Algorithm
SP	Special Publication

TIG	Technical Implementation Guidance
TLV	Tag-Length-Value
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique IDentifier
TIG	Technical Implementation Guidance
VM	Virtual Machine

A.3 Notation

The sixteen 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'. A sequence of bytes may be enclosed in single quotation marks, for example 'A0 00 00 01 16' rather than given as a sequence of individual bytes, 'A0' '00' '00' '01' '16'.

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

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

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

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

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

Appendix B—References

[1] Federal Information Processing Standard 201-1, Change Notice 1, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, March 2006. (See http://csrc.nist.gov)

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

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

[4] 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.
(See http://fips201ep.cio.gov/documents/TIG_SCEPACS_v2.2.pdf)

[5] NIST Special Publication 800-87, *Codes for the Identification of Federal and Federally-Assisted Organizations*, April 2008. (See http://csrc.nist.gov)

[6] *PKI for Machine Readable Travel Documents Offering ICC Read-Only Access Version - 1.1* Date - October 01, 2004. Published by authority of the Secretary General, International Civil Aviation Organization.

[7] *DoD CAC Middleware Requirements Release 3.0*, Version 1.0, Access Card Office, March 21, 2006. (See http://www.smart.gov/iab/documents/DoDcacMiddlewareRequirements.pdf)

[8] *DoD Implementation Guide for CAC Next Generation (NG)*, Version 2.6, DMDC Card Technologies & Identity Solutions Division (CTIS), November, 2006.
(See http://www.smart.gov/iab/documents/CACngImplementationGuide.pdf)

[9] *Personal Identity Verification Interoperability For Non-Federal Issuers*, May 2009.
(See http://www.idmanagement.gov)

[10] IETF RFC 4122, "A Universally Unique IDentifier (UUID) URN Namespace," July 2005.

[11] IETF RFC 4530, "Lightweight Directory Access Protocol (LDAP) entryUUID Operational Attribute," June 2006.

[12] IETF RFC 5280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008.

[13] Federal Information Processing Standard 180-3, *Secure Hash Standard (SHS)*, October 2008.
(See http://csrc.nist.gov)