NIST Special Publication 800-85A-4

PIV Card Application and Middleware Interface Test Guidelines (SP 800-73-4 Compliance)

David Cooper
Hildegard Ferraiolo
Ramaswamy Chandramouli
Jason Mohler

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-85A-4

C O M P U T E R   S E C U R I T Y

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

NIST Special Publication 800-85A-4

PIV Card Application and Middleware Interface Test Guidelines (SP 800-73-4 Compliance)

David Cooper
Hildegard Ferraiolo
Ramaswamy Chandramouli
Computer Security Division
Information Technology Laboratory

Jason Mohler
Electrosoft Services, Inc.
Reston, Virginia

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-85A-4

April 2016

Authority

National Institute of Standards and Technology Special Publication 800-85A-4
Natl. Inst. Stand. Technol. Spec. Publ. 800-85A-4, 153 pages (April 2016)
CODEN: NSPUE2

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-85A-4

Comments on this publication may be submitted to:

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

Reports on Computer Systems Technology

Abstract

NIST Special Publication (SP) 800-73 contains the technical specifications to interface with the smart card to retrieve and use the Personal Identity Verification (PIV) identity credentials. This document, SP 800-85A, contains the test assertions and test procedures for testing smart card middleware as well as the card application. The tests reflect the design goals of interoperability and PIV Card functions.

Keywords

application programming interface (API); authentication; card command interface; derived test requirements (DTR); FIPS 201; identity credential; middleware; Personal Identity Verification (PIV); smart cards; test assertions

Acknowledgements

The authors (David Cooper, Hildegard Ferraiolo Ramaswamy Chandramouli of NIST, and Jason Mohler of Electrosoft Services, Inc.) wish to thank their colleagues who reviewed drafts of this document and contributed to its development. 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 accuracy of this publication.

ALIGNING REVISION NUMBERS

WHAT HAPPENED TO SPECIAL PUBLICATION 800-85A REVISION 3 (SP 800-85A-3)?

Revision numbers between NIST Special Publications 800-73 and 800-85A were misaligned from the start because the initial publication of SP 800-85A did not occur until after the publication of SP 800-73 Revision 1 (SP 800-73-1). This resulted in versions of the two documents corresponding as follows: SP 800-73-1 with SP 800-85A; SP 800-73-2 with SP 800-85A-1 (i.e., Revision 1); and SP 800-73-3 with SP 800-85A-2.

This revision numbering mismatch created ongoing uncertainty and confusion regarding which revision of SP 800-85A was consistent with which revision of SP 800-73. To reduce this uncertainty going forward, revision number 3 has been skipped for SP 800-85A, and the present version of SP 800-85A has been given revision number 4 (SP 800-85A-4) since this version is consistent with the updates to SP 800-73 Revision 4 (SP 800-73-4). Future revisions of SPs 800-73 and 800-85A will maintain the revision number consistency.

Table of Contents

List of Figures

List of Tables

1. Introduction

Federal Information Processing Standard 201, *Personal Identity Verification (PIV) of Federal Employees and Contractors* [FIPS 201], was developed to establish standards for identity credentials. FIPS 201 sets the minimum requirements for a federal personal identification system that meets the control and security objectives of Homeland Security Presidential Directive (HSPD) 12 [HSPD 12]. FIPS 201 also gives the technical specifications of components and processes required for the interoperability of PIV Cards[1] with the access control and PIV card management systems throughout the Federal Government. FIPS 201 is accompanied by three documents:

+ NIST Special Publication 800-73-4 (NIST SP 800-73-4) [SP 800-73] specifies interface requirements for retrieving and using the identity credentials from the PIV Card. It also defines a PIV data model, which details the structure and format of the information stored on the PIV Card.
+ NIST SP 800-76-2 [SP 800-76] contains technical specifications for biometric data mandated in FIPS 201.
+ NIST SP 800-78-4 [SP 800-78] specifies the cryptographic algorithms and key sizes for performing cryptographic operations on PIV data objects defined as part of the PIV data model.

This test guidance document specifies the test plan, processes, derived test requirements, and the detailed test assertions/conformance tests for testing the following PIV software components:

+ PIV Middleware (implements PIV Client API).
+ PIV Card Application.

The present document, SP 800-85A-4, supersedes SP 800-85A-2 in its entirety, effective immediately upon publication.

1.1 Purpose

The objective of this document is to provide test requirements and test assertions that could be used to validate the compliance/conformance of two PIV components – *PIV Middleware* and *PIV Card Applications* – with the specifications in NIST SP 800-73-4. Because NIST SP 800-73-4 specifications were developed for meeting interoperability goals of FIPS 201, the conformance tests in this document provide the assurance that the set of PIV Middleware and PIV Card Applications that have passed these tests are interoperable. This in turn facilitates procurement of FIPS 201-conformant products that meet the goals of HSPD-12.

1.2 Scope

This document provides guidelines for running conformance tests for the following three classes of specifications in NIST SP 800-73-4:

[1] The term PIV Card in the context of this document refers to a smart card loaded with a PIV Card Application.

+ PIV Data Objects Representation (Section 4, Part 1 of SP 800-73-4) and Data Types and Their Representation (Section 5, Part 1 of SP 800-73-4).
+ PIV Card Application Card Command Interface (Part 2 of SP 800-73-4).
+ PIV Client Application Programming Interface (Part 3 of SP 800-73-4).

The functions specified in the Client API are to be supported by PIV Middleware. The commands specified in the PIV Card Application Card Command Interface are to be supported by PIV Card Applications, with appropriate security conditions for executing each command and for accessing/storing each of the data objects associated with the application. The overall design of the commands has to be based on the concepts outlined in NIST SP 800-73-4 Part 2, Section 2 - Concepts and Constructs. The presence of mandatory data objects on the PIV Card has to be verified. The data objects associated with PIV Card Application have to be tested for their accessibility and storage using the specified identifiers. Thus, the three classes of specifications listed above span the following two main PIV components: PIV Middleware and PIV Card Application. Hence the test suite provided in this document consists of the following two broad categories of tests:

+ PIV Middleware tests.
+ PIV Card Application tests.

The above tests are developed through the following two-step process:

+ **Derived Test Requirements (DTR).** These are constructed from the 'shall' statements in SP 800-73-4 specifications.
+ **Test Assertions.** These provide the tests that need to be performed to test each of the requirements under DTRs as well as tests with appropriate execution conditions for each of the commands in the interface to realize the associated return/response status codes specified in SP 800-73-4 Part 2.

This document does not provide conformance tests for any other software used in the PIV system such as the back-end access control software, card issuance software, card reader/biometric reader drivers, and specialized service provider software such as cryptographic service provider modules and biometric service provider modules. This document does not address nor provide conformance tests for SP 800-76-2.

1.3 Target Audience

This document is intended to:

+ Enable developers of PIV Middleware and PIV Card Applications to develop their software modules to be testable for interface requirements specified in SP 800-73-4.
+ Enable developers of PIV Middleware and PIV Card Applications to develop self-tests as part of the development effort.
+ Enable testing laboratories authorized to perform conformance tests on PIV Middleware and PIV Card Applications to develop tests that cover the test suite provided in this document.

1.4 Document Overview

The document is organized as follows:

+ Section 2 provides a conceptual software overview of a typical PIV system and introduces the PIV test components.
+ Section 3 lists the various elements of the test suite under the two broad categories of tests (PIV Middleware tests and PIV Card Application tests) provided in this document.
+ Section 4 provides an overview of the DTR construction process.
+ Section 5 gives a brief description of the test assertion for each of the three specification classes covered by this document (refer to Section 1.3).
+ Section 6 explains the documentation required from both the component owners and test labs for conducting the testing process.
+ Section 7 details the acceptance criteria for each type of test.
+ Section 8 explains the test compliance process and failure review.
+ Appendix A includes DTRs based on specifications in SP 800-73-4.
+ Appendix B includes client application programming interface (API) test assertions.
+ Appendix C includes PIV Card command interface test assertions.
+ Appendix D contains a list of acronyms used in the document.
+ Appendix E contains the list of documents used as references by this document.

2. System Overview

The conceptual architecture involving the PIV Middleware and PIV Card Application for which conformance tests are given in this document is shown in Figure 1. The conformance tests in this document apply to the areas highlighted with dashed lines in Figure 1.

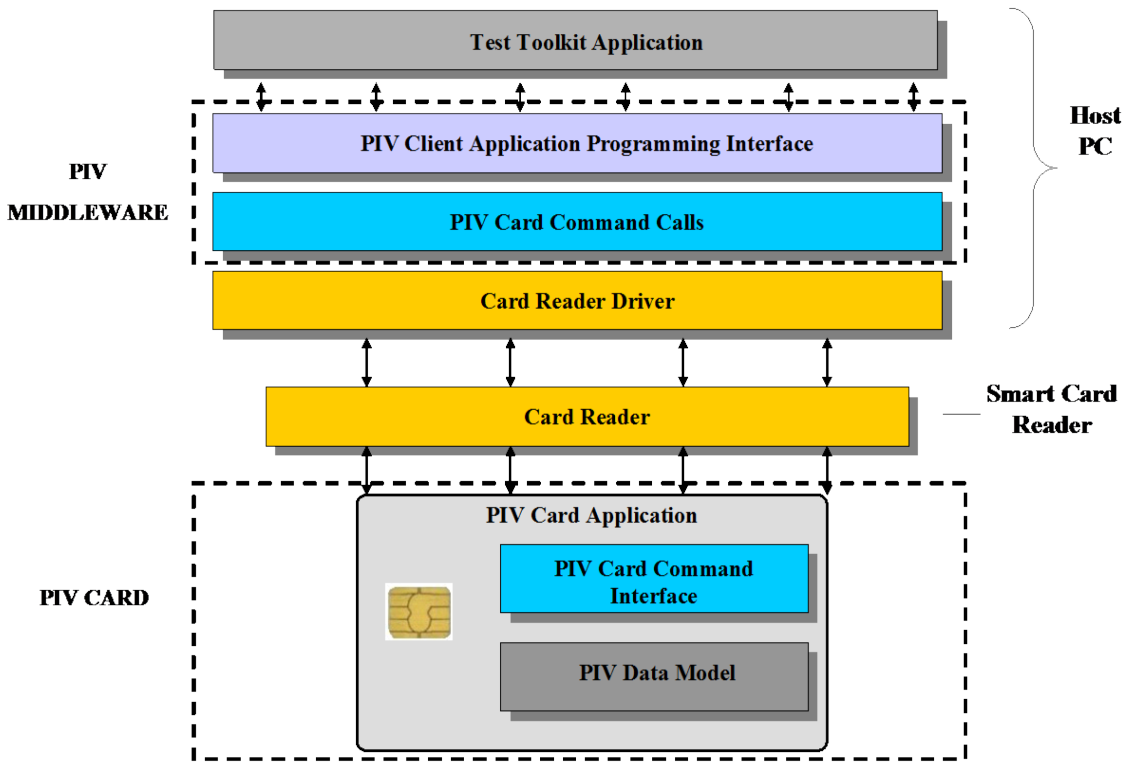

Figure 1: PIV Conformance Test Architecture

+ PIV Middleware is a software application that is the interface between an agency's PIV implementation and the PIV Card Application. It allows the agency's applications to remain independent of the underlying operating system platform. The PIV Middleware has the following two functions:

    1. It implements the functions for the PIV Client API (Part 3 of SP 800-73-4).

    2. It generates the appropriate commands (also called application protocol data units or APDUs) for the PIV Card Command Interface (card edge interface – Part 2 of SP 800-73-4) and thus communicates with the PIV Card Application.

  The PIV Card Application resides on the card, implements the commands in the PIV Card Command Interface (Part 2 of SP 800-73-4), and provides access to objects of the PIV data model. The PIV data model defines the logical use of the on-card application space including the SP 800-73-4 Part 1 required data objects and data elements, along with the size and structure of each object.

2.1 Test Plan

The test plan identifies the tasks/artifacts required for testing the PIV Middleware and PIV Card Applications. These artifacts include the following: PIV Middleware and a smart card populated with a PIV Card Application; the test toolkit (or test scripts), which implements the test assertions; and the various infrastructure devices needed to interface with the card and the card reader. The components involved in the test plan and the elements of the test configuration for the two broad categories of tests presented in this document are discussed in the next two subsections.

2.2 Test Set-up

The test system consists of the following components:[2]

- Test toolkit application software that resides on a personal computer (PC).
- Smart card (SC) readers:
  - An ISO/IEC 7816 and PC/SC-compliant contact-based smart card reader and
  - An ISO/IEC 14443 and PC/SC-compliant contactless smart card reader.

  or

  - A dual interface reader.
- A personal identification number (PIN) pad or a keyboard that can transmit the PIN to the smart card reader.
- A set of test PIV Cards, loaded with PIV Card Application, with a contact interface that is compliant with ISO/IEC 7816 and a contactless interface that is compliant with ISO/IEC 14443, or a test PIV Card emulator.
- PIV Middleware application.

These components will be used in different configurations based on the type of test being conducted in the test bed.

2.3 Test System Configuration

The test system shown in Figure 2 will be configured in both the PIV Middleware tests and the PIV Card Application tests to accommodate the different components to be tested, as explained in Section 3.

[2] Compliance of the readers and input devices with an external standard such as ISO/IEC 7816 is not addressed in this document.

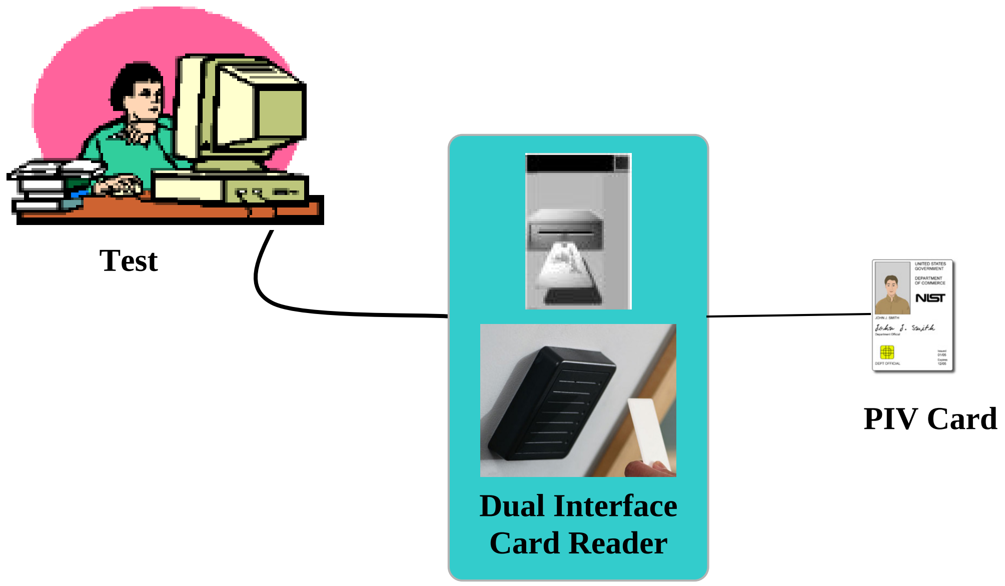

Figure 2: Test System Configuration

2.3.1 PIV Middleware Test Configuration

The middleware test configuration is used to test a vendor's middleware software application that implements the PIV client API and generates the appropriate commands in the PIV card command interface (refer to Table A-1 for mapping between the client API and card command interface). The middleware test configuration is depicted in Figure 3.

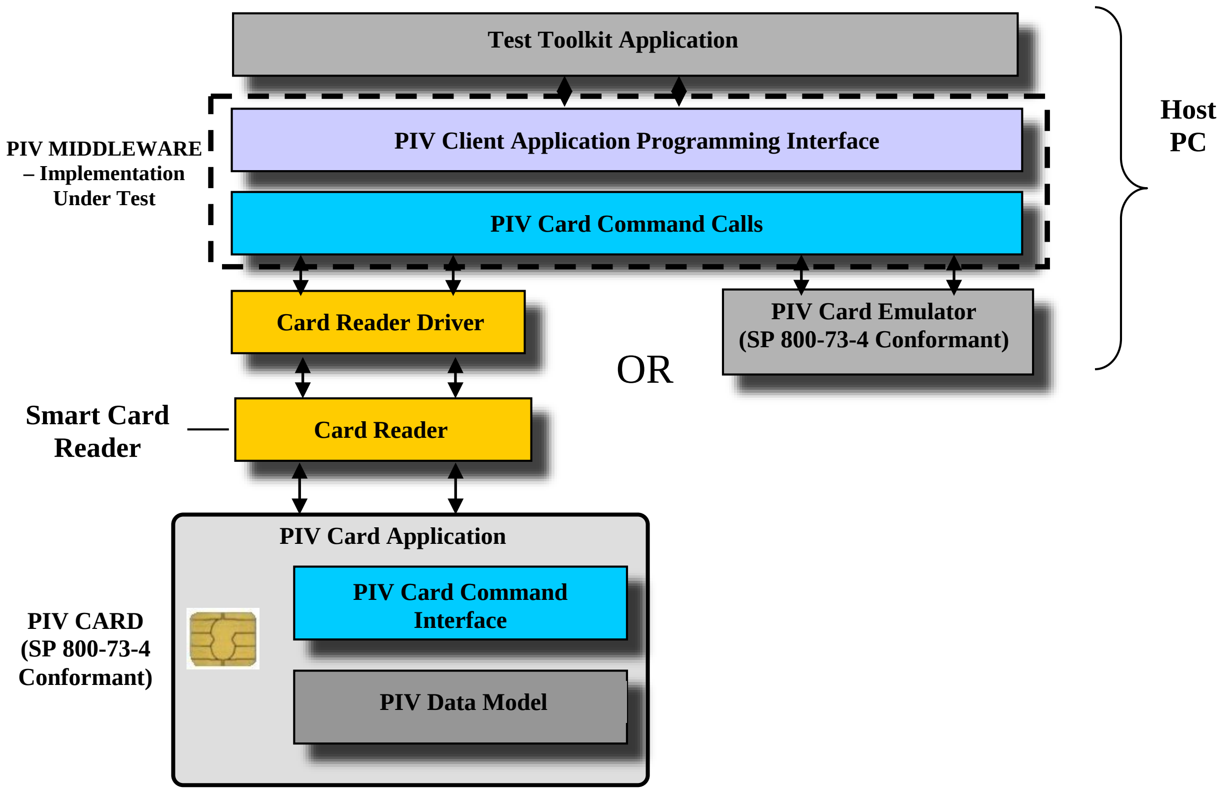
Test Toolkit Application
PIV MIDDLEWARE – Implementation Under Test
PIV Client Application Programming Interface
PIV Card Command Calls
Host PC
Card Reader Driver
PIV Card Emulator (SP 800-73-4 Conformant)
OR
Smart Card Reader
Card Reader
PIV Card Application
PIV CARD (SP 800-73-4 Conformant)
PIV Card Command Interface
PIV Data Model

Figure 3: Middleware Test Configuration

The following list shows the test system components included in this configuration:

+ Test toolkit application software.
+ Vendor provided PIV Middleware, which is the subject of this test (also called the implementation under test or IUT) and

One of the following combinations:

+ Contact and contactless smart card readers or a dual interface reader together with
+ A PIN input mechanism together with
+ A dual interface FIPS 201 conformant PIV Card loaded with "SP 800-73-4 Part 2 conformant PIV Card Application" (Refer to Section 7.2 for definition).

or

+ A PIV card emulator that emulates the behavior of a PIV Card Application.

The test toolkit application software resides on the test computer and facilitates the execution and management of both test suites explained in Section 3. For the PIV Middleware Test, the test system (Figure 2) will be configured so that the vendor provided PIV Middleware under test is also installed on the test computer and interacts with the SP 800-73-4 conformant test cards via the card reader(s).

2.3.2 PIV Card Application Test Configuration

The card application test configuration is used to test any PIV Card Application through commands of the PIV card command interface defined in SP 800-73-4 Part 2. The following list shows the test system components included in this configuration:

+ Test toolkit application software.
+ Contact and contactless smart card readers or a dual interface reader.
+ A PIN input mechanism.
+ A PIV Card loaded with a PIV Card Application that supports contact and contactless interfaces and is the subject of this test (also called implementation under test or IUT).

For the PIV Card Application Test, the test system shown in Figure 2 will be configured such that the test toolkit application software directly interacts with the PIV Card under test via the card reader(s). The PIV Card Application Test configuration is depicted in Figure 4.

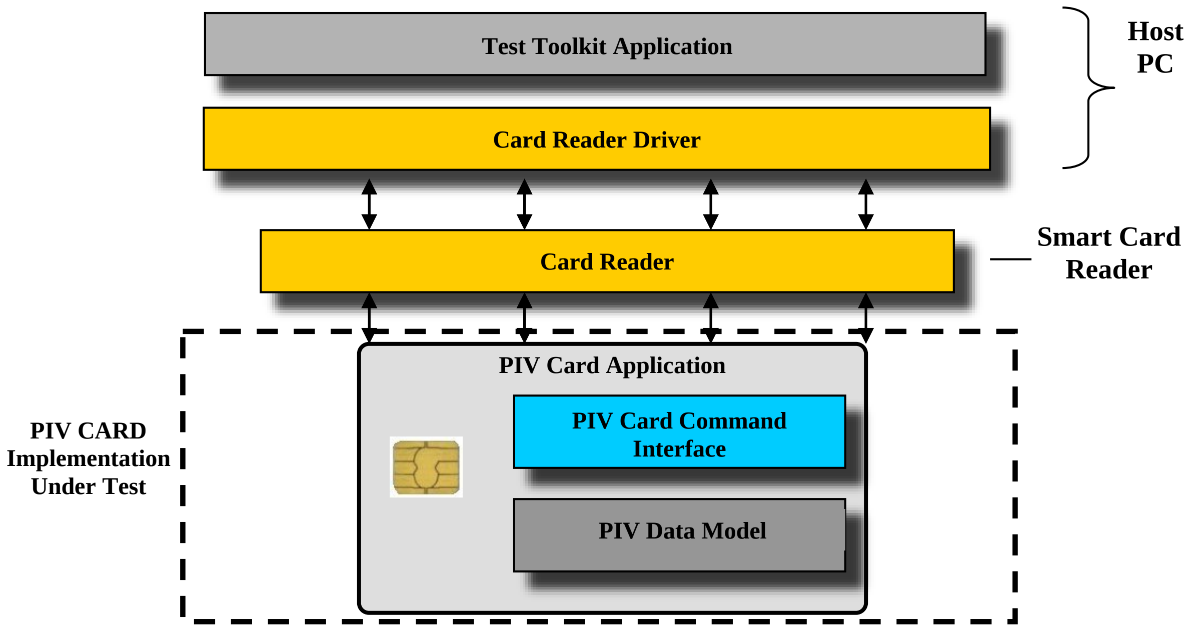

Figure 4: PIV Card Application Test Configuration

3. Test Suite Elements

Based on the conceptual software architecture shown in Figure 1, the PIV software components that are subject to testing are as follows:

+ PIV Middleware that implements the functions for the PIV Client API and interfaces with the PIV Card Application (resident on the card) by generating commands (APDUs) to the PIV card command interface.
+ PIV Card Application that implements the PIV Card Application card command interface, accesses and modifies the content of PIV data objects, and facilitates realization of PIV authentication use cases.

3.1 PIV Middleware Tests

These tests will validate that the PIV Middleware conforms to the specification in Part 3 of SP 800-73-4. Conformance criteria include correct implementation of the functions for the PIV client API, generation of appropriate commands for the PIV card command interface to communicate with the PIV Card Application, and return of the prescribed response codes to the calling agency application. This test, however, does not validate the functional requirements or the testing of the FIPS 201-mandated card application parameters, which are covered under the PIV data model tests as specified in SP 800-85B.

The following PIV Middleware functions are tested for conformance of the PIV Middleware without support for secure messaging (SM) and the virtual contact interface (VCI):

1. pivConnect
2. pivDisconnect
3. pivSelectCardApplication
4. pivLogIntoCardApplication
5. pivGetData
6. pivLogoutOfCardApplication
7. pivCrypt
8. pivPutData
9. pivGenerateKeyPair
10. pivMiddlewareVersion

For PIV Middleware with support for SM and VCI, the following additional PIV Middleware function is tested for conformance:

11. pivEstablishSecureMessaging

These functions will be tested for their response to both the valid and the error conditions as defined by this document. To conduct these tests, a smart card with an "SP 800-73-4-conformant PIV Card Application" (refer to Section 7.2 for definition) must be accessible.

3.2 PIV Card Application Tests

PIV Card Application tests cover the following:

+ The PIV Card Application card command interface as per Part 2 of SP 800-73-4, including the security conditions for executing each command in the interface as well as the security conditions for accessing and storing each of the associated data objects.
+ Presence of all mandatory data objects as well as accessibility and storage of all implemented data objects using the identifiers specified in Part 1 of SP 800-73-4.

The tests are performed through test scripts communicating directly with a PIV Card through the API of the driver that comes with the card reader.

3.2.1 PIV Card Application Card Command Interface Tests

These tests will validate that the card under test can successfully execute the commands in the PIV card command interface. Successful execution constitutes the card responding with appropriate data and response status codes to the commands sent by the test system. It also involves setting state variables within the PIV Card. For example, the criteria for successful execution of the SELECT command involve the following:

+ The response status code returned is '90 00'.
+ The application property template is returned with the correct format and content.
+ The “PIV Card Application” is the value of “currently selected application” (state variable) on the card.

The card command interface test suite includes conformance tests for the following PIV Card Application commands:

+ Data access commands.
  - SELECT
  - GET DATA
+ Card authentication commands.
  - VERIFY
  - CHANGE REFERENCE DATA
  - RESET RETRY COUNTER
  - GENERAL AUTHENTICATE
+ Credential initialization and administration commands.
  - PUT DATA
  - GENERATE ASYMMETRIC KEY PAIR

The card edge commands will be validated against the following conditions:

+ Card interface type (contact vs. contactless, including virtual contact interface).

+ Precondition for use (PIN, OCC verified, cryptographic authentication).
+ Expected response status codes.
+ Appropriate state variables set in the card.

3.2.2 PIV Data Objects Accessibility and Storage Tests

The testing covers the following data objects:

+ The seven mandatory data objects as defined in Part 1 of SP 800-73-4:
    - Card Capability Container.
    - Card Holder Unique Identifier (CHUID).
    - X.509 Certificate for PIV authentication.
    - X.509 Certificate for Card Authentication.
    - Cardholder Fingerprints for off card comparison.
    - Cardholder Facial Image.
    - Security Object.
+ The two data objects that are mandatory if the cardholder has a government-issued email account at the time of credential issuance:
    - X.509 Certificate for Digital Signature.
    - X.509 Certificate for Key Management.
+ The twenty-seven optional data objects, also defined in Part 1 of SP 800-73-4:
    - Printed Information.
    - Discovery Object.
    - Key History Object.
    - 20 retired X.509 Certificates for Key Management.
    - Cardholder Iris Images.
    - Pairing Code Reference Data Container.
    - Secure Messaging Certificate Signer.
    - Biometric Information Templates Group Template.

The data objects will be validated for the following conditions:

+ Presence of all mandatory data objects and those optional objects in the vendor documentation.
+ Accessibility and storage of data objects using the appropriate BER-TLV tags (specified identifiers – Section 4, Part 1 of SP 800-73-4).
+ Appropriate container size allocations for each of the data objects.
+ Data objects access rule (e.g., PIN vs. no PIN).

+ Security condition for data objects storage (cryptographic authentication).
+ Appropriate card interface type for accessing each of the data objects (contact vs. contactless vs. secure messaging vs. virtual contact interface).

4. Derived Test Requirements

DTRs show the type of tests required based on the specifications in SP 800-73-4. These specifications cover expected command behavior (in the case of interface specification), data object representation (in the case of PIV data model) and data contents (in the case of PIV authentication use cases).

Each DTR consists of the following:

+ Actual condition statements taken/derived from the SP 800-73-4 specification – these include conditions for successful command execution for each command as well as exception behaviors explicitly called out through 'shall' statements in SP 800-73-4. Those exception behaviors that are implicit in SP 800-73-4 through listing of error codes associated with each command are tested only through Test Assertions (Appendices B, C and D) and are not part of the DTR condition statements. The condition statements are identified by codes starting with 'AS' followed by a running sequence that denotes the section in this document where they occur. All DTRs that are new to Revision 4 of this document are identified by the suffix '-R4'. Updated DTRs that existed previously are requirements updated in SP 800-73-4 and these retain their original identifiers.
+ Required Vendor Information – these include information that the vendors are mandated to provide in their documentation. The required vendor information is identified by codes starting with 'VE' followed by a running sequence that denotes the section in this document where they occur. All vendor information requirements that are new to Revision 4 of this document are identified by the suffix '-R4'. Updated required vendor information that existed previously are requirements updated in SP 800-73-4 and these retain their original identifiers.
+ Required Test Procedures – these are actions that the tester has to perform in order to satisfy the requirements stated in actual condition statements. These include verifying the information mandated in the "Required Vendor Information" for the condition as well as performing software-based tests. It must be mentioned, however, that some of the required test procedures will not be called out explicitly for verification of information in the associated "Required Vendor Information." In these instances, it is implicitly assumed that the information is provided by the vendor and verified by the tester. The Required Test Procedures have identifiers starting with 'TE' followed by a running sequence that denotes the section in this document where they occur. All test procedure requirements that are new to Revision 4 of this document are identified by the suffix '-R4'. Updated required test procedures that existed previously are requirements updated in SP 800-73-4 and these retain their original identifiers.

Validations of some DTRs are not covered by the test assertions provided in this document. These DTRs require compliance of the component with an external specification or standard such as ISO/IEC 7816 or ISO/IEC 14443. No required test procedures are provided for these DTRs, and a note is added to indicate that the assertion is externally tested. The tester checks the vendor documentation for claimed compliance with such requirement or the presence of an external test/compliance certificate obtained from the related standards testing body, when applicable.

Some DTRs cannot be validated through the test tools provided in this document. For example, the test tool cannot access the asymmetric private keys generated and stored on the card. Therefore, a note is added to indicate the assertion is not separately tested for these DTRs. The same note is added for DTRs that make general statements on the nature of the PIV Card and are validated as a result of the validation of many other DTRs. For example, the statement “[e]ach command that appears on the card command interface shall be implemented by a *card application* that is resident on the ICC [integrated circuit card]” is validated through the entire card command interface test and does not require an individual test assertion.

5. Test Assertions

Test assertions are statements of behavior, action, or condition that can be measured or tested. They provide the procedures to guide the tester in executing and managing the test. They include the purpose of the test, starting conditions and prerequisites, success criteria, and post-test conditions, when applicable. A list of test assertions can be seen in Appendices B and C.

The following three sets of test assertions are included in this document:

+ PIV client API test assertions (see Section 3.1 for overview).
+ PIV card command interface test assertions (per Section 3.2.1).
+ PIV data objects accessibility and storage test assertions (per Section 3.2.2).

An overview of each of the above classes of test assertions is given in Sections 5.2 through 5.4.

5.1 Mapping from Test Categories to Test Assertions

All the DTRs in Appendix A conceptually come under one of the two broad categories of tests stated in Section 3, i.e., PIV Middleware tests and PIV Card Application tests. Similarly, each test assertion makes specific references to the related sections in SP 800-73-4 or the related DTRs. However, overall there is a many-to-many mapping from the test suite elements (individual tests) under each of these two broad categories of tests to the DTRs (i.e., one test can map to many DTRs and one DTR can map to many tests). A similar type of relationship exists between DTRs and test assertions. To narrow the search space for cross references, Table 5-1 presents a cross-referencing guide showing the relevant DTR sections (with the section in SP 800-73-4 from which they were derived) and test assertion sections with respect to test classes in the two broad categories of tests.

Table 5-1: Cross-referencing Guide

Category/Classes of Test	DTR Section(s)	Test Assertion Section(s)
(1a) PIV Card Application Tests—PIV Card Application Card Command Interface Tests (Section 3.2.1)	▪ Appendix A.1: Concepts and Constructs (Section 2, SP 800-73-4 Part 2) ▪ Appendix A.5: PIV Card Application Card Command Interface (Section 3 and 4, SP 800-73-4 Part 2)	Appendix C—PIV Card Command Interface Test Assertions
(1b) PIV Card Application Tests—PIV Data Object s Accessibility and Storage Tests (Section 3.2.2)	▪ Appendix A.2: PIV Data Objects Representation (Section 4, SP 800-73-4 Part 1) ▪ Appendix A.3: Data Types and Their Representation (Section 5, SP 800-73-4 Part 1, SP 800-73-4 Part 2)	Appendix C—PIV Data Objects Accessibility and Storage Test Assertions
(2) PIV Middleware Tests (Section 3.1)	▪ Appendix A.1: Concepts and Constructs (Section 2, SP 800-73-4 Part 2) ▪ Appendix A.2: PIV Data Objects Representation (Section 4, SP 800-73-4 Part 1)	Appendix B—PIV Client API Test Assertions

	▪ Appendix A.3: Data Types and Their Representation (Section 5, SP 800-73-4 Part 1, SP 800-73-4 Part 2) ▪ Appendix A.4: PIV Client API (SP 800-73-4 Part 3)	

5.2 PIV Client API Test Assertions

This section provides conformance tests in the form of test assertions for the functions specified in Section 3, SP 800-73-4 Part 3 (referred to as the client API) that the PIV Middleware is expected to support. The test assertions are described through a test assertions template. The template provides placeholders for describing the purpose of the test, the preconditions required to exercise the test, the parameter values used in test invocation, and the expected results as well as the state of the PIV system (value of state variables), if any, that will be affected by the test run (post-condition).

The conformance tests are run against the PIV Middleware, which in turn interacts with the PIV Card Application resident on the PIV Card. Hence, there are two pieces of software (PIV Middleware and PIV Card Application) that determine the outcome of each test run. Because the focus of the tests is the behavior of the PIV Middleware, the test configuration assumes the presence of a validated PIV Card Application.

The test assertions derived from SP 800-73-4 and SP 800-78-4 are demonstrated by test cases in Appendix B.

The PIV client API test cases are based on the following assumptions:

+ There is a PIV Card with a validated PIV Card Application.
+ A valid connection description is provided for the PIV Card Application.
+ A valid physical connection exists between an instance of the PIV card reader and the host where the PIV Middleware resides.
+ No other application is currently connected to the PIV Card Application.

5.3 PIV Card Command Interface Test Assertions

This section provides conformance tests in the form of test assertions for the command set that is specified in Section 3, Part 2 of SP 800-73-4 (PIV Card Application Card Command Interface) that the PIV Card Application is required to support. The test assertions are described through a test assertions template. The template provides placeholders for describing the purpose of the test, the preconditions required to exercise the test, the parameter values used in test invocation, and the expected results as well as the state of the PIV system (value of state variables), if any, that will be affected by the test run (post-condition).

The conformance tests are run to validate the PIV Card Application. Interaction with the PIV Card Application takes place through the API of the driver that comes with the card reader.

The test assertions derived from SP 800-73-4 Part 2 are demonstrated in test cases in Appendix C.

The following assumptions have been made with regard to the PIV Card command interface test cases:

+ The PIV Card being tested (IUT) is inserted into the contact reader or placed near a contactless reader.
+ A valid PC/SC connection exists between the test system and an instance of the reader.
+ No application is currently connected to the PIV Card Application.
+ No other contactless card is within the proximity of the contactless reader.

5.4 PIV Data Objects Accessibility and Storage Test Assertions

The following assumptions have been made with respect to the PIV data object representation test assertions:

+ A PIV Card Application with a valid Application Identifier (AID) is resident on the card.
+ The PIV Card Application is expected to have implemented all seven mandatory PIV data objects of the PIV data model on the card.
+ The PIV Card Application is expected to have implemented both conditionally mandatory PIV data objects[3] of the PIV data model on the card.
+ The presence of any one or more of the twenty-seven optional PIV data objects on the PIV Card is known from the vendor documentation.

[3] The two data objects that are conditionally mandatory, that is mandatory only if the cardholder has a government-issued email account at the time of credential issuance, are the X.509 Certificate for Digital Signature and the X.509 Certificate for Key Management.

6. Test and Compliance Documentation

There are two sets of compliance documentation: vendor required and test facility generated.

The vendor-required documents consist of the following:

+ **Installation and Execution instructions (for PIV Middleware):** The vendor provides technical instructions and other documentation to aid the testing personnel in installing and using the PIV Middleware implementation under test. The PIV Middleware implementation could be in any high-level programming language. Since all the implementations have to be tested from a common test program, the PIV Middleware vendor submitting the product for testing may have to provide Java wrapper programs in some cases to the test facility. The purpose of the wrapper program is to translate the test execution calls made using the test program to the PIV Middleware implementation's native program calls.
+ **Technical documentation (for both PIV Card Application and PIV Middleware):** The vendor-supplied technical documentation must include the detailed technical description and the design of the implementation to be tested. This document includes, at a minimum, all the required vendor information specified in DTRs in Appendix A of this document.
+ **Security-related information:** The following security related information shall be provided by the vendor: (a) PIV Card Application PIN, (b) PIN Unblocking Key (PUK), (c) Global PIN, (d) pairing code, (e) cryptographic algorithms supported by the card, (f) minutia data, and (g) the number of unsuccessful attempts using (1) wrong PIV Card Application PIN, (2) wrong Global PIN, (3) wrong PUK, (4) and wrong minutia data.

The test facility-generated documents are required for performing and reporting the test process. The following are some of the examples:

+ **Checklists:** Checklists provide the tester with a list of actions and requirements to complete before the test starts. Information required in the preconditions section of the assertions is included in the checklists.
+ **Test logs:** A test log is kept for each test run on any component and is used to summarize the results of all the tests run.
+ **Test reports:** These provide the background (environmental information) for each of the test cases as well as summary of outcomes from test runs (from test logs) associated with each test case.

A test case is a sequence of command/function invocations that pertain to a given execution condition for the 'command/function under test'. For example, if the GET DATA command is the command/function under test, then the execution condition 'Invocation of this function after PIN verification' will consist of the following sequence of command/function invocations – SELECT, VERIFY, GET DATA, and collectively constitutes a test case. There may be many test runs for this test case. The function invocations returning the expected return codes for a test case in all test runs indicate that the command/function has been implemented correctly.

7. Acceptance Criteria

Acceptance criteria are based on the compliance of the item under the test with the requirements defined in FIPS 201 and the accompanying special publication documents. The criteria are further specified in the following sections, based on the type of test being conducted.

7.1 Acceptance Criteria for the PIV Middleware Test

The PIV Middleware test acceptance criteria will be based on the middleware application under the test passing the SP 800-73-4 client API test assertions. The middleware should return appropriate return codes in response to executing the client API functions as defined in Section 3, Part 3 of SP 800-73-4. The middleware should also be able to send the correct card commands to and interpret the responses received from the "SP 800-73-4 conformant PIV Card Application" (refer to Section 7.2 for definition). The test assertions detail the pass/fail criteria defined for each test case that is designed to test a certain condition being tested.

PIV Middleware that supports the SP 800-73-4 client API with SM will be required to test against both the SP 800-73-4 client API test assertions and the SP 800-73-4 client API with SM test assertions. All SP 800-73-4 client API with SM test assertions are to be conducted over a contactless interface.

7.2 Acceptance Criteria for the PIV Card Application Tests

Acceptance criteria for the PIV Card Application tests are based on the PIV Card Application passing the following two classes of tests: PIV Card Application card command interface tests and PIV data objects accessibility and storage tests. The PIV Card Application that has passed these classes of tests is called an "SP 800-73-4 conformant PIV Card Application."

For PIV Card Application card command tests, the PIV Card Application should send the appropriate response status codes and application data in response to commands. It should also set or reset certain card state variables and thus fulfill the test postconditions.

For the PIV data objects accessibility and storage tests, the PIV Card Application should show the presence of all mandatory PIV data objects and published optional PIV data objects. It should also demonstrate the ability to access and store all the above data objects using the correct BER-TLV tag under the appropriate security conditions and interfaces (contact, contactless, secure messaging, or VCI) and that the containers for storing them satisfy the specified minimum size requirements.

The acceptance criteria for the testing of PIV Card functionalities, for which FIPS 201 makes reference to external documents (such as digital signature formats), is based on visual verification of vendor-provided documents and test/compliance certificates.

8. Test and Compliance Process

The PIV software component that passes all applicable tests, as explained in this document, will be considered conformant. This document provides the technical details for the testing of the two PIV software components. In this context, compliance means:

+ Passing the related test assertions explained in this document, and
+ Passing the inspection/verification of the required vendor documentation.

The certified and/or accredited test laboratory that will conduct the testing has the following responsibilities:

+ Prepare and provide the test application forms and the documentation,
+ Receive and configure the PIV software component to be tested,
+ Conduct the test with the testing toolkit,
+ Review the test results and report failures,
+ Inspect the vendor documentation, and
+ Communicate the results.

Upon vendor’s submission of the request for PIV component certification, the required documentation, and the PIV software components to be tested, the test laboratory configures the test system, records all preconditions, and runs the applicable suite of tests for the submitted PIV component. After conducting the tests, the test laboratory evaluates the test results and communicates the Test Results Summary (TRS) and Test Run Details (TRD) to the vendor.

The Test Results Summary provides the overall environmental information (date and time the tests were conducted, the tester name, vendor product identifier, etc.) as well as the summary conclusion for tests associated with that particular class. The format of the summary report will vary depending upon the test classes. The TRS associated with each of the three classes are:

+ PIV Client API Test Summary.
+ Card Command Interface Test Summary.
+ PIV Data Objects Accessibility and Storage Test Summary.

The TRDs are used to log the details of each test run associated with each of the three classes in the test suite. They provide the details of the outcome of each test run for various execution conditions. This detailed report will enable the product vendor to make the necessary logic changes to the implementation of the various commands/interfaces and data object representations in order to become fully conformant.

8.1 Failure Review

The test will be repeated once for components that do not pass the tests. After the retest, the tester prepares, for each failure, a discrepancy report that summarizes the purpose of the test, the progression of steps, and the responses received from the tested components. The discrepancy report will be internally reviewed and discussed by the test lab before an official response is sent to the vendor. Vendors who object to the results presented in the discrepancy report must explain their reason for the objection. If the reason necessitates another retest, the test laboratory may consider repeating the test. Otherwise, the test lab will seek the guidance of the NIST personnel on the failure before the component is returned to the vendor to be corrected.

Appendix A—Derived Test Requirements

All DTRs that are new to Revision 4 of this document are new requirements introduced in SP 800-73-4. These are referenced with the suffix -R4 (e.g., AS0*X*-R4). Updated DTRs that existed previously are requirements updated in SP 800-73-4 and these retain their original identifiers.

A.1 Concepts and Constructs

A.1.1 Platform Requirements

AS01.01: The PIV Card Application shall place the following requirements 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.**

Note: This assertion is not separately tested.

A.1.2 Card Applications

AS01.02: Each command that appears on the card command interface shall be implemented by a card application that is resident on the ICC.

Note: This assertion is not separately tested – collection of DTRs for all commands implicitly tests this assertion.

AS01.03: Each card application shall have a globally unique name called its Application Identifier (AID) [ISO/IEC 7816, Part 4].

Note: This assertion is tested as part of AS05.05 through AS05.10.

AS01.04: 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.

Note: This assertion is tested as part of AS05.11.

AS01.05: The Proprietary Identifier eXtension (PIX) of the AID shall contain an encoding of the version of the card application.

Note: This assertion is tested as part of the **AS05.05** through **AS05.10**.

A.1.2.1 Personal Identity Verification Card Application

AS01.06: The AID of the PIV Card Application shall be: 'A0 00 00 03 08 00 00 10 00 01 00'.

Note: This assertion is tested as part of the AS05.05 through AS05.10.

AS01.07: The AID of the PIV Card Application shall consist of the NIST Registered application provider IDentifier (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.

Note: This assertion is tested as part of the AS05.05 through AS05.10.

A.1.2.2 Default Selected Card Application

AS01.08: 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.

Required Vendor Information

VE01.08.01: The vendor shall specify in its documentation the default selected card application.

Required Test Procedures

TE01.08.01: The tester shall review the vendor's documentation and validate that there is a default selected card application, which matches with the one specified by the vendor in VE01.08.01.

A.1.3 Security Architecture

A.1.3.1 Access Control Rule

AS01.09: An access control rule shall consist of an access mode and a security condition.

Note: This assertion is not separately tested.

AS01.10: 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.

Note: This assertion is not separately tested.

AS01.11: 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.

Note: This assertion is not separately tested.

A.1.3.2 Security Status

AS01.12: Associated with each authenticable entity shall be a set of one or more Boolean variables, each called a security status indicator of the authenticable entity.

Note: The security status indicators are tested indirectly through the functional testing.

AS01.13: The security status indicator of an authenticable entity shall be TRUE if the credentials associated with the security status indicator of the authenticable entity have been authenticated and FALSE otherwise.

Note: The security status indicators are tested indirectly through the functional testing.

AS01.14: A successful execution of an authentication protocol shall set the security status indicator associated with the credential used in the protocol to TRUE.

Note: The security status indicators are tested indirectly through the functional testing.

AS01.14A-R4: 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.

Note: The security status indicators are tested indirectly through the functional testing.

AS01.15: 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.

Note: This assertion is not separately tested.

AS01.16: 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.

Required Vendor Information

VE01.16.01: The vendor shall specify in its documentation that the application security status indicators are set to FALSE when the currently selected application changes from one application to another.

Required Test Procedures

TE01.16.01: The tester shall review the vendor's documentation and validate that it contains the requirement stated in VE01.16.01.

AS01.16A-R4: The security status indicators associated with the PIV Card Application PIN, the PIN Unblocking Key (PUK), OCC, pairing code, and the PIV Card Application

Administration Key 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.

Required Vendor Information

VE01.16A-R4.01: The vendor shall specify in its documentation that the security status indicators associated with the PIV Card Application PIN, the PIN Unblocking Key (PUK), OCC, pairing code, and the PIV Card Application Administration Key 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.

Required Test Procedures

TE01.16A-R4.01: The tester shall review the vendor's documentation and validate that it contains the requirement stated in VE01.16A-R4.01.

A.1.3.3 Authentication of an Individual

AS01.17: The pairing code shall be exactly 8 bytes in length and the PIV Card Application PIN shall be between 6 and 8 bytes in length.

Note: This assertion is tested as part of AS05.22A.

AS01.18: If the actual length of PIV Card Application PIN is less than 8 bytes it shall be padded to 8 bytes with 'FF' when presented to the card command interface. The 'FF' padding bytes shall be appended to the actual value of the PIN.

Note: This assertion is tested as part of AS05.22A.

AS01.18A-R4: The bytes comprising the PIV Card Application PIN and pairing code shall be limited to values 0x30 – 0x39, the ASCII values for the decimal digits '0' – '9'.

Note: This assertion is tested as part of AS05.22A.

AS01.18B-R4: The PIV Card Application shall enforce the minimum length requirement of six bytes for the PIV Card Application PIN (i.e., shall verify that at least the first six bytes of the value presented to the card command interface are in the range 0x30 – 0x39) as well as the other formatting requirements specified in Section 2.4.3 of Part 2 of SP 800-73-4.

Note: This assertion is tested as part of AS05.22A.

AS01.18C-R4: The PUK shall be 8 bytes in length, and may be any 8-byte binary value. That is, the bytes comprising the PUK may have any value 0x00 – 0xFF.

Note: This assertion is tested as part of AS05.22A.

AS01.18D-R4: If the Global PIN is used by the PIV Card Application, then the above encoding, length, padding, and enforcement of minimum PIN length requirements for the PIV Card Application PIN shall apply to the Global PIN.

Note: This assertion is tested as part of AS05.22A.

A.1.4 Status of PIV Card Application

AS01.19: The state of the PIV Card Application shall be as follows, when the PIV Card Application is the currently selected application:

1. **The "global security status" indicator shall always be defined. It can be used by all applications on the card platform and is maintained by PIV Platform.**
2. **The "currently selected application" shall always be defined. 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. The "currently selected application" is maintained by the PIV Platform.**
3. **The "application security status" indicator shall always be defined. These indicators are local to the PIV Card Application and are maintained by the PIV Card Application.**

Note: This assertion is not separately tested.

A.1.5 Card Platform Configuration

AS01.20: Both single-chip/dual-interface and dual-chip implementations are acceptable.

Note: This assertion is not separately tested.

AS01.21: In the single-chip/dual-interface configuration, the PIV Card Application shall be provided the information regarding which interface is in use.

Required Vendor Information

VE01.21.01: The card operating system should inform the PIV Card Application the communication interface in use.

Required Test Procedures

TE01.21.01: The tester shall validate that the card platform informs the PIV Card Application of the interface being used.

Note: This assertion is not separately tested. This assertion is indirectly tested by verifying whether the card application returns an appropriate error response code for those commands that cannot be exercised through contactless interface. The tester shall verify an appropriate response code is returned.

TE01.21.02: The tester shall validate that the PIV Card Application checks that a contact interface is being used for contact-only APDUs.

Note: This assertion is not separately tested. This assertion is indirectly tested by verifying whether the card application returns an appropriate error response code for those commands that cannot be exercised through contactless interface when VCI is not in use and the commands are not contact only commands.

AS01.22: In the dual-chip configuration, a separate PIV Card Application shall be loaded on each chip.

Note: This assertion is not separately tested.

A.2 PIV Data Model

A.2.1 PIV Card Data Objects

AS02.01: A PIV Card Application shall contain seven mandatory interoperable data objects, two conditionally mandatory data objects (if the cardholder has a government-issued email account at time of credential issuance) and twenty-seven optional data objects for interoperable use.

- **The seven mandatory data objects are the following: 1. Card Capability Container; 2. Card Holder Unique Identifier; 3. X.509 Certificate for PIV Authentication; 4. X.509 Certificate for Card Authentication; 5. Cardholder Fingerprints; 6. Cardholder Facial Image; and 7. Security Object**

- **The two conditionally mandatory data objects are the following: 1. X.509 Certificate for Digital Signature; and 2. X.509 Certificate for Key Management**.

- **The twenty-seven optional data objects for interoperable use are the following: 1. Printed Information; 2. Discovery Object; 3. Key History Object; 4. 20 retired X.509 Certificates for Key Management; 5. Cardholder Iris Images; 6. Biometric Information Templates Group Template; 7. Secure Messaging Certificate Signer; and 8. Pairing Code Reference Data Container.**

Note: This assertion is not separately tested.

A.2.2 OIDs and Tags of PIV Card Application Data Objects

AS02.02: For the purpose of constructing PIV Card Application data object names in the CardApplicationURL in the Card Capability Container 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'.

Note: This assertion is not separately tested as it is being deprecated.

AS02.03: For all data objects present on the card, the object identifiers (OIDs) used by PIV Middleware to refer to them, and associated BER-TLV tags used by PIV Card Application command interface shall conform to the entries in Table 3, Part 1 of SP 800-73-4.

Required Vendor Information

VE02.03.01: The vendor shall state in its documentation the list of all the data objects present on the card along with the BER-TLV tags associated with them.

VE02.03.01A: The vendor shall state in its documentation the list of all OIDs used by the PIV Middleware to refer to PIV data objects and the associated BER-TLV tags to which they refer.

Required Test Procedures

TE02.03.01: The tester shall validate that the BER-TLV tags of all the data objects present on the card conform to the Table 3, Part 1 of SP 800-73-4, and accurately represent the actual data objects observed by the tester as being implemented on the card.

TE02.03.01A: The tester shall validate that all of the OIDs in Table 3, Part 1 of SP 800-73-4 can be used to read PIV data objects from a PIV Card using the pivGetData function.

A.3 Data Types and Their Representations

A.3.1 PIV Algorithm Identifier

AS03.01: The algorithm identifiers for the cryptographic algorithms implemented on the card shall conform to entries in Table 6-2 of SP 800-78-4.

Required Vendor Information

VE03.01.01: The vendor shall state in its documentation the identifiers associated with all the algorithms supported by the card.

Required Test Procedures

TE03.01.01: The tester shall review the vendor's documentation and validate the algorithm identifiers implemented on the card as being compliant with Tables 6-2 and 6-3 of SP 800-78-4.

A.3.2 Application Property Template

AS03.02: Upon selection, the PIV Card Application shall return the application property template described in Table 3, Part 2 of SP 800-73-4.

Required Test Procedures

TE03.02.01: The tester shall validate that the information returned upon selection of the PIV application is in conformance with Table 3, Part 2 of SP 800-73-4.

A.3.3 Authenticator

AS03.03: The authenticator BER-TLV used on the PIV client application programming interface shall have the structure described in Table 3, Part 3 of SP 800-73-4.

Required Vendor Information

VE03.03.01: The vendor shall provide in its documentation a list of all the authenticators along with their tags and possible values, when applicable.

Required Test Procedures

TE03.03.01: The tester shall review and validate the vendor's documentation to ensure that it states the correct tags for the "Reference Data" and "Key Reference" as specified in Table 3, Part 3 of SP 800-73-4.

A.3.4 Connection Description

AS03.04: Moved to Appendix A.4.1.1 (AS04.02A-R4).

AS03.05: Withdrawn

A.3.5 Key References

AS03.06: The key reference, when represented as a byte, occupies bits b8 and b5-b1, while b7 and b6 shall be set to 0.

Note: This assertion is not separately tested.

AS03.07: The key references used on all PIV interfaces shall be from the list found in Tables 4a and 4b, Part 1 of SP 800-73-4 and SP 800-78-4, Table 6-1.

Note: This assertion is not separately tested.

AS03.08: Withdrawn

A.3.6 WITHDRAWN - Status Words

A.3.7 OCC Data

AS03.09-R4: If OCC is implemented, the export of the biometric reference data shall not be allowed.

Required Vendor Information

VE03.09-R4.01: The vendor shall state in its documentation that the exportation of the biometric reference data is not allowed by the card.

Required Test Procedures

TE03.09-R4.01: The tester shall review the vendor’s documentation and validate that the exportation of the biometric reference data is not allowed by the card.

A.4 Client Application Programming Interface

A.4.1 Entry Points for Communication

AS04.01: Entry points on the PIV client API shall include all functions listed in Table 1, Part 3 of SP 800-73-4 for middleware that supports secure messaging. For middleware that does not support secure messaging entry points on the PIV client API shall include all functions listed in Table 1, Part 3 of SP 800-73-4 with the exception of pivEstablishSecureMessaging.

Note: This assertion is tested as part of AS04.02 through AS04.10A-R4.

Required Vendor Information & Required Test Procedures

To test the entry points or commands that are supported by the PIV Middleware the only information that the vendor has to provide is the PIV Middleware version. All parameter values for exercising the commands have to be obtained from the vendor documentation, using the mapping of PIV Middleware functions to the PIV Card Application card commands that are listed in Table A-1 below. Hence this section contains only tester requirements in terms of Required Test Procedures.

Table A-1: PIV Command Mapping

PIV MiddlewarePIV Middleware Functions	Section		PIV Card Application Card Command[4]	Mapping Description
pivConnect	A.4.1.1		No equivalent command	For establishing a connection session with the card reader.
pivDisconnect	A.4.1.2		No equivalent command	For disconnecting a connection session with the card reader.

[4] It is assumed that some of these functions will use GET RESPONSE and chaining to accomplish the read or write to the card.

PIV MiddlewarePIV Middleware Functions	Section		PIV Card Application Card Command [4]	Mapping Description
pivSelectCardApplication	A.4.2.1		SELECT	Passes the AID value. Sets the value for 'Currently Selected Application' on the PIV Card. Establishes the PIV Card Application security status.
pivLogIntoCardApplication	A.4.2.2		VERIFY	Provides the key reference for PIV Card Application PIN, Global PIN, pairng code, or OCC as well as its corresponding value. Sets/updates the PIV Card Application security status on the card; optionally (according to discovery object) establishes VCI with pairing code after pivEstablishSecureMessaging is invoked.
pivLogOutOfCardApplication	A.4.2.4		VERIFY	Resets the security status on all local key references.
pivGetData	A.4.2.3		GET DATA	Maps the OID to BER-TLV tag for the selected object.
pivPutData	A.4.4.1		PUT DATA	Maps the OID to BER-TLV tag for the selected object.
pivGenerateKeyPair	A.4.4.2		GENERATE ASYMMETRIC KEY PAIR	Passes the key reference and cryptographic mechanism identifier value.
pivCrypt	A.4.3.1		GENERAL AUTHENTICATE	Passes the key reference, cryptographic algorithm reference and the string to be acted upon. Sets/updates the PIV Card Application security status on the card, if applicable.
pivMiddlewareVersion	A4.1.3		No equivalent command	Returns the PIV Middleware version supported by the PIV Middleware IUT.
pivEstablishSecureMessaging	A.4.2.5		GENERAL AUTHENTICATE [5]	Passes the key reference and cryptographic algorithm reference to the PIV Card Application in order to establish secure messaging. Establishes, controls, and maintains the session key.

A.4.1.1 pivConnect

AS04.02: The purpose of pivConnect is to connect the PIV API to the PIV Card Application on a specific ICC.

[5] In order to establish a VCI, pivLogIntoCardApplication may be required (as indicated in the Discovery Object) after successful execution of pivEstablishSecureMessaging.

TE04.02.01: The tester shall validate that the PIV Middleware implements pivConnect as per SP 800-73-4, Part 3.

AS04.02A-R4: The connection description BER-TLV used on the PIV client API shall have the structure described in Table 2, Part 3 of SP 800-73-4.

Required Vendor Information

VE04.02A-R4.01: The vendor shall provide in its documentation the format and content of the connection description templates implemented.

Required Test Procedures

TE04.02A-R4.01: The tester shall review the vendor's documentation to confirm the presence of the information provided in VE04.02A-R4.01 and that the connection description template conforms to Table 2, Part 3 of SP 800-73-4.

A.4.1.2 pivDisconnect

AS04.03: The purpose of pivDisconnect is to disconnect the PIV API from the PIV Card Application and the ICC containing the PIV Card Application.

Required Test Procedures

TE04.03.01: The tester shall validate that the PIV Middleware implements pivDisconnect as per SP 800-73-4 Part 3.

AS04.03A-R4: If secure messaging has been established then the PIV Middleware shall zeroize the secure messaging session keys when the pivDisconnect command is sent.

Required Vendor Information

VE04.03A-R4.01: The vendor shall specify in its documentation that the PIV Middleware zeroizes the secure messaging session keys as a part of the implementation of pivDisconnect.

Required Test Procedures

TE04.03A-R4.01: The tester shall review the vendor's documentation and validate that it asserts that the PIV Middleware zeroizes the secure messaging session keys as part of the implementation of pivDisconnect.

A.4.1.3 pivMiddlewareVersion

AS04.03B-R4: PIV Middleware that returns a versionString of "800-73-4 Client API with SM" shall implement all PIV Middleware functions listed in Table 1 of SP 800-73-4 Part 3 and

be able to recognize and process all mandatory and optional PIV data objects. PIV Middleware that returns a versionString of "800-73-4 Client API" shall implement all PIV Middleware functions listed in Table 1 except pivEstablishSecureMessaging and shall be able to recognize and process all mandatory and optional PIV data objects. The pivMiddlewareVersion's purpose is to return the PIV Middleware version string. For SP 800-73-4 Part 3 conformant PIV Middleware, the parameter returns "800-73-4 Client API" or "800-73-4 Client API with SM" if optional secure messaging is supported.

Required Test Procedures

TE04.03B-R4.01: The tester shall validate that the PIV Middleware supports all functions listed in Table 1 of SP 800-73-4 Part 3 and implements the pivMiddlewareVersion as per SP 800-73-4 Part 3 by returning the appropriate parameter string "800-73-4 Client API" parameter string or "800-73-4 Client API with SM" if optional secure messaging is supported.

A.4.2 Entry Points for Data Access

A.4.2.1 pivSelectCardApplication

AS04.04: The purpose of pivSelectCardApplication is to set the PIV Card Application as the currently selected card application and establish the PIV Card Application's security state.

Required Test Procedures

TE04.04.01: The tester shall validate that the PIV Middleware implements pivSelectCardApplication as per SP 800-73-4, Part 3.

AS04.04A-R4: If the length of application properties is longer than the buffer allocated by the PIV client application, then the PIV Middleware shall return PIV_INSUFFICIENT_BUFFER, but shall still set APLength to the length of the application properties.

Required Test Procedures

TE04.04A-R4.01: The tester shall ensure that the PIV Middleware returns PIV_INSUFFICIENT_BUFFER if the length of application properties is longer than the buffer allocated by the client application.

A.4.2.2 pivLogIntoCardApplication

AS04.05: The purpose of pivLogIntoCardApplication is to set the security state within the PIV Card Application.

Required Test Procedures

TE04.05.01: The tester shall validate that the PIV Middleware implements the pivLogIntoCardApplication as per SP 800-73-4 Part 3.

AS04.05A-R4: The PIV Middleware shall not submit authenticators to the PIV Card over a contactless interface without secure messaging. If secure messaging has not been established, then the pivLogIntoCardApplication function shall return PIV_SECURITY_CONDITIONS_NOT_SATISFIED.

Required Vendor Information

VE04.05A-R4: The vendor shall state in its documentation that the card supports the assertions made in AS04.05A-R4.

Required Test Procedures

TE04.05A-R4.01: The tester shall validate that that the vendor documentation contains the information required in VE04.05A-R4.

A.4.2.3 pivGetData

AS04.06: The purpose of pivGetData is to return the entire data content of the named data object.

Required Test Procedures

TE04.06.01: The tester shall validate that the PIV Middleware implements the pivGetData as per SP 800-73-4 Part 3.

AS04.06A-R4: If the length of the retrieved data is longer than the buffer allocated by the client application, then the PIV Middleware shall return PIV_INSUFFICIENT_BUFFER, but shall still set DataLength to the length of the retrieved data.

Required Test Procedures

TE04.06A-R4.01: The tester shall ensure that the PIV Middleware returns PIV_INSUFFICIENT_BUFFER if the length of the retrieved data is longer than the buffer allocated by the client application

A.4.2.4 pivLogoutOfCardApplication

AS04.07: The purpose of pivLogoutOfCardApplication is to reset the application security state/status of the PIV Card Application.

Required Test Procedures

TE04.07.01: The tester shall validate that the PIV Middleware implements the pivLogoutOfCardApplication as per SP 800-73-4 Part 3.

A.4.2.5 pivEstablishSecureMessaging

AS04.07A-R4: The purpose of pivEstablishSecureMessaging is to establish secure messaging with the PIV Card Application.[6]

Required Test Procedures

TE04.07A-R4.01: The tester shall validate that the PIV Middleware implements the pivEstablishSecureMessaging as per SP 800-73-4 Part 3.

AS04.07B-R4: After successful execution of the key establishment protocol, the PIV Middleware shall perform all subsequent GET DATA, VERIFY, and GENERAL AUTHENTICATE commands over secure messaging, with the exception of any subsequent uses of the GENERAL AUTHENTICATE command to perform the key establishment protocol.

Required Test Procedures

TE04.07B-R4.01: The tester shall validate that upon successful execution of the key establishment protocol the GET DATA, VERIFY, and GENERAL AUTHENTICATE commands are only sent to the card using secure messaging when the pivGetData, pivLogIntoCardApplication, and pivCrypt middleware functions are called.

AS04.07C-R4: The session keys established after successful execution of the key establishment protocol in Section 4.1 of SP 800-73-4 Part 2 shall be zeroized in the following circumstances: (i) the card is reset; (ii) an error occurs in secure messaging; or (iii) new session keys are requested by the client application by sending a GENERAL AUTHENTICATE command to the card to perform the key establishment protocol using the PIV Secure Messaging key.

Required Vendor Information

VE04.07C-R4.01: The vendor shall state in its documentation that the card supports the assertions made in AS04.07C-R4.

Required Test Procedures

TE04.07C-R4.01: The tester shall validate that the vendor documentation contains the information required in VE04.07C-R4.01.

A.4.3 Entry Points for Cryptographic Operations

A.4.3.1 pivCrypt

[6] The PIV Middleware maintains the session keys and performs the cryptographic operations for secure messaging.

AS04.08: The purpose of pivCrypt is to perform a cryptographic operation such as encryption or signing on a sequence of bytes.[7]

Required Test Procedures

TE04.08.01: The tester shall validate that the PIV Middleware implements the pivCrypt as per SP 800-73-4 Part 3.

AS04.08A-R4: If the value of keyReference is '04' (PIV Secure Messaging key) then the PIV Middleware shall return PIV_INVALID_KEYREF_OR_ALGORITHM.

Note: This assertion is tested as part of AS04.08.

AS04.08B-R4: If the length of the algorithm output is longer than the buffer allocated by the client application, then the PIV Middleware shall return PIV_INSUFFICIENT_BUFFER, but shall still set outputLength to the length of the algorithm output.

Required Test Procedures

TE04.08B-R4.01: The tester shall ensure that the PIV Middleware returns PIV_INSUFFICIENT_BUFFER if the length of the algorithm output is longer than the buffer allocated by the client application.

A.4.4 Entry Points for Credential Initialization and Administration

A.4.4.1 pivPutData

AS04.09: The purpose of pivPutData is to replace the entire data content of the named data object with the provided data.

Required Test Procedures

TE04.09.01: The tester shall validate that the PIV Middleware implements the pivPutData as per SP 800-73-4 Part 3.

AS04.09A-R4: The PIV Middleware shall not submit data provided to the pivPutData function over the contactless interface. If the PIV Middleware is not communicating with the PIV Card via the card's contact interface then the pivPutData function shall return PIV_FUNCTION_NOT_SUPPORTED.

Required Vendor Information

[7] 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, except SM on the client side, are performed outside the PIV Middleware.

VE04.09A-R4: The vendor shall state in its documentation that the card supports the assertions made in AS04.09A-R4.

Required Test Procedures

TE04.09A-R4.01: The tester shall validate that that the vendor documentation contains the information required in VE04.09A-R4.

A.4.4.2 pivGenerateKeyPair

AS04.10: The purpose of pivGenerateKeyPair is to generate an asymmetric key pair in the currently selected card application.

Required Test Procedures

TE04.10.01: The tester shall validate that the PIV Middleware implements the pivGenerateKeyPair as per SP 800-73-4 Part 3.

AS04.10A-R4: If the length of public key related data retrieved from the PIV Card is longer than the buffer allocated by the client application, then the PIV Middleware shall return PIV_INSUFFICIENT_BUFFER, but shall still set KeyLength to the length of the public key related data retrieved from the PIV Card.

Required Test Procedures

TE04.10A-R4.01: The tester shall ensure that the PIV Middleware returns PIV_INSUFFICIENT_BUFFER if the length of public key related data retrieved from the PIV Card is longer than the buffer allocated by the client application.

AS04.11-R4: The PIV Middleware shall not submit data provided to the pivGenerateKeyPair function over the contactless interface. If the PIV Middleware is not communicating with the PIV Card via the card's contact interface then the pivGenerateKeyPair function shall return PIV_FUNCTION_NOT_SUPPORTED.

Required Vendor Information

VE04.11-R4.01: The vendor shall state in its documentation that the card supports the assertions made in AS04.11-R4.01.

Required Test Procedures

TE04.11-R4.01: The tester shall validate that that the vendor documentation contains the information required in VE04.11-R4.01.

A.4.4.3 pivMiddlewareVersion

The Derived Test Requirements, and their associated required test procedures, for the pivMiddlewareVersion client application command have been moved to Appendix A.4.1.3.

A.5 PIV Card Application Card Command Interface

AS05.01: All PIV Card Application card commands listed in Table 2, Part 2 of SP 800-73-4 shall be supported by a PIV Card Application.

Required Vendor Information

VE05.01.01: The vendor shall provide the list of all PIV Card Application card commands, along with the interface(s) (contact or contactless, SM, VCI) they support, the security condition(s) they are subject to and their support for command chaining as implemented by the card.

Required Test Procedures

TE05.01.01: The tester shall review the vendor's documentation and validate that the information presented in response to VE05.01.01 by the vendor complies with Table 2, Part 2 of SP 800-73-4.

TE05.01.02: The tester shall validate that the card implements all the commands as required in Table 2, Part 2 of SP 800-73-4.

TE05.01.03: The tester shall validate that the commands are implemented only through the interfaces allowed as shown in Table 2, Part 2 of SP 800-73-4.

TE05.01.04: The tester shall validate that the commands are performed only if the security conditions associated with them are satisfied, as shown in the table, via the specified interface.

TE05.01.05: The tester shall validate that the commands as indicated in the table are allowed for chaining via the interface supported after the security condition is satisfied.

AS05.02: Card commands indicated with a 'Yes' in the Command Chaining column of Table 2, Part 2 of SP 800-73-4 shall support command chaining for transmitting a data string too long for a single command as defined in ISO/IEC 7816-4 [6].

Note: This assertion is tested as part of AS05.01.

AS05.03: The PIV Card Application shall return the status word of '6A 81' (Function not supported) when it receives a card command on the contactless interface marked "No" in the Contactless Interface column in Table 2, Part 2 of SP 800-73-4. The PIV Card Application may return a different status word (e.g., '69 82') if the card command can be performed over the contactless interface in support of card management. The PIV Card Application will only perform the command in support of card management if the requirements specified in Section 2.9.2 of FIPS 201-2 are satisfied.

Note: This assertion is tested as part of AS05.01.

AS05.04: Cryptographic protocols using private/secret keys that require the "PIN" or "OCC" security condition shall only be used on the contactless interface after a Virtual Contact Interface (VCI) has been established.

Note: This assertion is tested as part of AS05.01.

A.5.1 PIV Card Application Card Commands for Data Access

A.5.1.1 SELECT Card Command

AS05.05: The PIV Card Application shall be selected by providing its application identifier, 'A0 00 00 03 08 00 00 10 00 10 00', in the data field of the SELECT command.

Required Vendor Information

VE05.05.01: The vendor shall specify in its documentation the PIV Card Application Identifier.

Required Test Procedures

TE05.05.01: The tester shall validate that the PIV Card Application is selected by providing its application identifier as specified in AS05.05.

AS05.06: There shall be at most one PIV Card Application on any ICC.

Required Vendor Information

VE05.06.01: The vendor shall state in its documentation that there is only one PIV Card Application on the ICC.

Required Test Procedures

TE05.06.01: The tester shall review and validate the vendor's documentation as stated in VE05.06.01.

AS05.07: The PIV Card Application can also be made the currently selected application by providing a right-truncated version – that is, without the two-byte version number, '10 00' – in the data field of the SELECT command 'A0 00 00 03 08 00 00 10 00'

Required Vendor Information

VE05.07.01: The vendor shall provide the list of valid AIDs that the card supports and the mechanism(s) implemented to select the PIV Card Application.

Required Test Procedures

TE05.07.01: The tester shall review and validate that the information provided in VE05.07.01.

TE05.07.02: The tester shall validate that the PIV application is selectable by the right-truncated AID in the SELECT command.

AS05.08: The complete AID, including the two-byte version, of the PIV Card Application that became the currently selected application upon successful execution of the SELECT command (using the full or right-truncated PIV AID) shall be returned in the application property template.

Note: This assertion is tested as part of AS03.02.

AS05.09: If the currently selected application is the PIV Card Application when the SELECT command is sent and the AID in the data field of the SELECT command is either the AID of the PIV Card Application or its 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.

Required Vendor Information

VE05.09.01: The vendor shall provide information in its documentation stating compliance as required by AS05.09.

Required Test Procedures

TE05.09.01: The tester shall validate that when the currently selected application is the PIV Card Application and the SELECT command is sent with an AID that is either the AID of the PIV Card Application or its right-truncated version, then the PIV Card Application continues to be the currently selected application and the setting of all security status indicators in the PIV Card Application remains unchanged.

AS05.10: If the currently selected application is the PIV Card Application when the SELECT command is sent 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 currently selected application and all PIV Card Application security status indicators shall remain unchanged.

Required Vendor Information

VE05.10.01: The vendor shall provide information in its documentation validating the compliance with the statement in AS05.10.

Required Test Procedures

TE05.10.01: The tester shall validate that when the currently selected application is the PIV Card Application and the SELECT command is sent with an AID that is not a valid AID supported by the card, then the PIV Card Application continues to be the currently selected application and the setting of all security status indicators in the PIV Card Application remains unchanged.

AS05.11: 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.

Required Vendor Information

VE05.11.01: The vendor shall provide information in its documentation validating the compliance with the statement in AS05.11.

Required Test Procedures

TE05.11.01: The tester shall validate that when the currently selected application is the PIV Card Application and the SELECT command is sent with a valid AID supported by the ICC that is different from the PIV Card Application AID (or its right-truncated version), then PIV Card Application is deselected and its security status indicators are set to FALSE.

AS05.11A-R4: A PIV Card Application may use a subset of the cryptographic algorithms defined in SP 800-78-4. Tag 0xAC encodes the cryptographic algorithms supported by the PIV Card Application. The encoding of tag 0xAC shall be as specified in Table 5, Part 2 of SP 800-73-4. Each instance of tag 0x80 shall encapsulate one algorithm. The presence of algorithm identifier '27' or '2E' indicates that the corresponding cipher suite is supported by the PIV Card Application for secure messaging and that the PIV Card Application possesses a PIV Secure Messaging key of the appropriate size for the specified cipher suite. Tag 0xAC shall be present and indicate algorithm identifier 0x27 or 0x2E (but not both) when the PIV Card Application supports secure Messaging.

Note: This assertion is tested as part of AS05.34.

A.5.1.2 GET DATA Card Command

AS05.12: The GET DATA card command retrieves the data content of the single data object whose tag is given in the data field.

Note: This assertion is tested as part of AS05.01.

AS05.12A: The GET DATA card command retrieves the data content of the data object only after the access rule associated with the data object (Appendix A, Table 7, Part 1 of SP 800-73-4) evaluates to TRUE.

Required Vendor Information

VE05.12A.01: The vendor shall specify in its documentation the access rule for each of the data objects or make a reference to Table 7 in Appendix A, Part 1 of SP 800-73-4.

Required Test Procedures

TE05.12A.01: For implementations without the Discovery Object or implementations with the Discovery Object implemented and Bit 6 of the first byte of the PIN Usage Policy set to zero, the tester shall validate that all data objects that require a PIN are only accessible after a successful validation of the PIV Card Application PIN through the VERIFY command.

TE05.12A.02: For implementations with the Discovery Object implemented and Bit 6 of the first byte of the PIN Usage Policy set to one: 1) the tester shall validate that all data objects are accessible after a successful VERIFY with the PIV Card Application PIN; and 2) the tester shall validate that all data objects that require a PIN are accessible after a successful VERIFY with the Global PIN.

TE05.12A.02A-R4: For implementations with the Discovery Object implemented and Bit 5 of the first byte of the PIN Usage Policy set to one, the tester shall validate that the appropriate data objects are accessible after a successful VERIFY with OCC.

TE05.12A.03: The tester shall validate that all data objects whose access rule is "Always" are accessible with or without PIV Card Application PIN validation, Global PIN validation (if implemented as indicated in the Discovery Object), or OCC validation (if implemented as indicated in the Discovery Object) using the permitted interface.

A.5.2 PIV Card Application Card Commands for Authentication

A.5.2.1 VERIFY Card Command

AS05.13: PIV Card Applications for which both the PIV Card Application PIN and the Global PIN satisfy the PIV ACRs for PIV data object access and command execution shall implement the Discovery Object with the PIN Usage Policy set to 0x60 zz, 0x6C zz, 0x70 zz, or 0x7C zz, where zz is either 0x10 or 0x20, and may optionally implement the Discovery Object with the PIN Usage Policy set to 0x68 zz or 0x78 zz, where zz is either 0x10 or 0x20.

Required Vendor Information

VE05.13.01: The vendor shall confirm that the PIV Card Application PIN can be used for PIV data object access and command execution. If the Global PIN (in addition to the PIV Card Application PIN) is used for data access and command execution while the PIV Card Application is the currently selected application, the vendor shall state in its documentation that the card supports the assertion made in AS05.13.

Required Test Procedures

TE05.13.01: The tester shall validate that the PIV Card Application PIN can be used for PIV data object access and command execution. The tester shall validate that when the Global PIN satisfies the PIV ACRs for PIV data object access and command execution then: 1) the Discovery Object is implemented with the PIN Usage Policy set to 0x60 zz, 0x68 zz, 0x6C zz, 0x70 zz, 0x78 zz, or 0x7C zz, where zz is set to either 0x10 or 0x20; and 2) the Global PIN can be used for PIV data object access and command execution.

AS05.13A-R4: PIV Card Applications for which OCC satisfies the PIV ACRs for PIV data object access and command execution shall implement the Discovery Object with the first byte of the PIN Usage Policy set to 0x50, 0x58, 0x5C, 0x70, 0x78, or 0x7C.

Required Vendor Information

VE05.13A-R4.01: If OCC (in addition to the PIV Card Application PIN, and possibly the Global PIN) is used for data access and command execution while the PIV Card Application is the currently selected application, the vendor shall state in its documentation that the card supports the assertion made in AS05.13A-R4.

Required Test Procedures

TE05.13A-R4.01: The tester shall validate that when OCC satisfies the PIV ACRs for PIV data object access and command execution then: 1) the Discovery Object is implemented with the first byte of the PIN Usage Policy set to 0x50, 0x58, 0x5C, 0x70, 0x78, or 0x7C; and 2) OCC can be used for PIV data object access and command execution.

AS05.13B-R4: PIV Card Applications that implement the VCI shall implement the Discovery Object with the first byte of the PIN Usage Policy set to 0x4C, 0x5C, 0x6C, or 0x7C, and may optionally also implement the Discovery Object with the first byte of the PIN Usage Policy set to 0x48, 0x58, 0x68, or 0x78.

Required Vendor Information

VE05.13B-R4.01: If the PIV Card Application implements the VCI, the vendor shall state in its documentation that the card supports the assertion made in AS05.13B-R4.

Required Test Procedures

TE05.13B-R4.01: The tester shall validate that PIV Card Applications that implement the VCI implement the Discovery Object with the first byte of the PIN Usage Policy set to 0x4C, 0x5C, 0x6C, or 0x7C (see Table 1 of **SP 800-73-4** Part 1).

AS05.14: Key reference '80' specific to the PIV Card Application (i.e., local key references) and, optionally, the Global PIN with key reference '00', the OCC data (key references '96' and '97'), and pairing code (key reference '98') are the only key references that may be verified by the PIV Card Application’s VERIFY command. The PIV Card Application may allow other key references to be verified by the PIV Card Application’s VERIFY command, if they are used for card management operations.

Required Vendor Information

VE05.14.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.14.

Required Test Procedures

TE05.14.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.14.01.

AS05.15: Key reference '80' shall be able to be verified by the PIV Card Application VERIFY command.

Note: This assertion is tested as part of AS05.13.

AS05.16: If the PIV Card Application contains the Discovery Object as described in Part 1 of SP 800-73-4 and Bit 6 of the first byte of the PIN Usage Policy value is one, then key reference '00' shall be able to be verified by the PIV Card Application VERIFY command.

Required Vendor Information

VE05.16.01: The vendor shall specify in its documentation if the Global PIN is implemented with the VERIFY command to satisfy access control rules to read PIN protected PIV data objects. If implemented, the vendor shall also specify the Discovery Object to be present on card with Bit 6 of the first byte of the PIN Usage Policy value set to one.

Required Test Procedures

TE05.16.01: The tester shall validate that if the PIV Card Application contains the Discovery Object and Bit 6 of the first byte of the PIN Usage Policy value is one, then key reference '00' is able to be verified by the PIV Card Application VERIFY command.

AS05.16A-R4: If the key reference is '98' and the authentication data in the command data field does not match the reference data associated with the key reference, the command shall fail and the PIV Card Application shall return the status word '63 00'. If the authentication data in the command data field does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the PIV Card Application may return the status word '6A 80' instead of '63 00'. If status word '6A 80' is returned, the security status of the key reference shall remain unchanged. If status word '63 00' is returned, the security status of the key reference shall be set to FALSE.

Required Test Procedures

TE05.16A-R4.01: The tester shall validate that the card implements AS05.16A-R4 as specified.

AS05.17: If the key reference is '00', '80', '96', or '97' and 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'. In order to protect against blocking over the contactless interface, PIV Card Applications that implement secure messaging shall define an issuer-specified intermediate retry value for each of these key references and return '69 83' if the command is submitted over the contactless interface (over secure messaging or the VCI, as required for the key reference) and the current value of the retry counter associated with the key reference is at or below the issuer-specified intermediate retry value. If status word '69 83' is returned, then the comparison shall not be made, and the security status and the retry counter of the key reference shall remain unchanged.

Required Vendor Information

VE05.17.01: The vendor shall specify in its documentation the reset value of the retry counters associated with all the key references implemented on the card.

Required Test Procedures

TE05.17.01: The tester shall validate that the PIV Card Application returns '69 83' in response to the VERIFY command when the retry counter associated with the key reference is zero.

AS05.18: The card command shall fail if the key reference is '00' or '80' and the VERIFY command is not submitted over either the contact interface or the VCI or if the key reference is '96', '97', or '98' and the VERIFY command is submitted over the contactless interface without secure messaging. In either case, the security status and the retry counter of the key reference shall remain unchanged.

Required Vendor Information

VE05.18.01: The vendor shall specify in its documentation the conditions (and associated status word) when the command will fail.

Required Test Procedures

TE05.18.01: The tester shall validate that if the key reference is '00' or '80' and the VERIFY command is not submitted over either the contact interface or the VCI, or if the key reference is '96', '97', or '98' and the VERIFY command is submitted over the contactless interface without secure messaging, then the card command fails and the security status and the retry counter of the key reference remain unchanged.

AS05.18A-R4: If the key reference is '96' or '97' and the authentication data in the command data field is not of length 3N, where N satisfies the requirements for minimum and maximum number of minutiae specified in the BIT, then the card command shall fail, and the PIV Card Application shall return the status word '6A 80'. The security status and the retry counter of the key reference shall remain unchanged.

Required Vendor Information

VE05.18A-R4.01: The vendor shall specify in its documentation that if the key reference is '96' or '97' and the authentication data in the command data field is not of length 3N, where N satisfies the requirements for minimum and maximum number of minutiae specified in the BIT, then the card command fails, the PIV Card Application returns the status word '6A 80', and the security status and the retry counter of the key reference remain unchanged.

Required Test Procedures

TE05.18A-R4.01: The tester shall validate that the card implements AS05.18A-R4 as specified.

AS05.19: If the key reference is '00', '80', '96', or '97' and the authentication data in the command data field is properly formatted and does not match reference data associated with the key reference, then the card command shall fail, the PIV Card Application shall return the status word '63 CX', 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.

Required Vendor Information

VE05.19.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.19.

Required Test Procedures

TE05.19.01: The tester shall validate that when the VERIFY command fails the retry counter associated with the key reference is decremented by one.

AS05.20: If the card command succeeds then the security status of the key reference shall be set to TRUE. If the key reference is '00', '80', '96', or '97' then the retry counter associated with the key reference shall be set to the reset retry value associated with the key reference.

Required Vendor Information

Note: This vendor information is reviewed as part of VE05.17.01.

Required Test Procedures

Note: This assertion is tested as part of AS05.17.

AS05.21: Moved requirement into AS05.19.

AS05.22A: If the PIN value in the reference data field of the command field is not padded to 8 bytes, the PIV Card Application shall return the status word '6A 80'.

Required Vendor Information

VE05.22A.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.22A.

Required Test Procedures

TE05.22A.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.22A.01 and the card returns status word '6A 80' when the PIN information in the reference data field of the command is not padded to 8 bytes.

AS05.22B: If the key reference is set to a value other than what is supported by the card, the PIV Card Application shall return the status word '6A 88' (key reference not found).

Required Vendor Information

VE05.22B.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.22B.

Required Test Procedures

TE05.22B.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.22B.01.

AS05.22C-R4: The VERIFY command shall reset the security status of the key reference in P2 when the P1 parameter is 'FF' and both L_c and the data field are absent. The security status of the key reference specified in P2 shall be set to FALSE and the retry counter associated with the key reference shall remain unchanged.

Required Vendor Information
VE05.22C-R4.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.22C-R4.

Required Test Procedures

TE05.22C-R4.01: The tester shall validate that when using the VERIFY command the security status of the key reference in P2 is reset when the P1 parameter is 'FF' and both L_c and the data field are absent. The tester shall also validate that in this scenario the retry counter remains unchanged.

AS05.22D-R4: If the key reference is '00' or '80' and the authentication data in the command data field does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the card command shall fail and the PIV Card Application shall return either the status word '6A 80' or '63 CX'. If status word '6A 80' is returned, the security status and the retry counter of the key reference shall remain unchanged. If status word '63 CX' is returned, 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.

Required Vendor Information

VE05.22D-R4.01: The vendor shall state in its documentation that the card supports the assertions made in AS05.22D-R4

Required Test Procedures

TE05.22D-R4.01: The tester shall validate that: 1) the vendor documentation contains the information required in VE05.22D-R4.01; and 2) the card returns status word '6A 80' or '63 CX' based on the conditions mentioned in AS05.22D-R4.

A.5.2.2 CHANGE REFERENCE DATA Card Command

AS05.23: 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. The PIV Card Application may allow the reference data associated with other key references to be changed by the PIV Card Application CHANGE REFERENCE DATA, if PIV Card Application will only perform the command with other key references if the requirements specified in Section 2.9.2 of FIPS 201-2 are satisfied.

Required Vendor Information

VE05.23.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.23.

Required Test Procedures

TE05.23.01: The tester shall validate that reference data associated with key reference '80' can be changed by the PIV Card Application's CHANGE REFERENCE DATA command. If the Discovery Object is implemented with Bit 6 of the first byte of the PIN Usage Policy set to one and the implementation supports changing the Global PIN with the CHANGE REFERENCE DATA command, then the tester shall also validate that key reference '00' reference data can be changed by the CHANGE REFERENCE DATA command. If the PUK can be changed with CHANGE REFERENCE DATA, the tester shall validate that reference data associated with key reference '81' can be changed by the PIV Card Application CHANGE REFERENCE DATA command.

AS05.24: WITHDRAWN

AS05.24A-R4: If key reference '81' is specified and the command is submitted over the contactless interface (including SM or VCI), then the card command shall fail. If key reference '00' or '80' is specified and the command is not submitted over either the contact interface or

the VCI, then the card command shall fail. In each case, the security status and the retry counter of the key reference shall remain unchanged.

Required Vendor Information

VE05.24A-R4.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.24A-R4.

Required Test Procedures

TE05.24A-R4.01: The tester shall validate that if the CHANGE REFERENCE DATA command is submitted over the contactless interface (including SM or VCI) with key reference '81', the PIV Card Application fails. The tester shall validate that if the CHANGE REFERENCE DATA command, with key reference '00' or '80', is submitted over the contactless interface, but not the VCI, then the card command fails. In both cases, the tester shall validate that the security status and the retry counter of the key reference remain unchanged.

AS05.25: 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' (Reference data change operation blocked). If the command is submitted over the contactless interface (VCI) and the current value of the retry counter associated with the key reference is at or below the issuer-specified intermediate retry value (see Section 3.2.1 of Part 2 of SP 800-73-4), 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'.

Required Vendor Information

VE05.25.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.25.

Required Test Procedures

TE05.25.01: The tester shall validate that when the current value of the retry counter associated with the key reference is zero, the reference data associated with the key reference does not change and the PIV Card Application returns '69 83' (Reference data change operation blocked). The tester shall validate that when the CHANGE REFERENCE DATA command is submitted over the VCI and the current value of the retry counter associated with the key reference is at or below the issuer-specified intermediate retry value, the reference data associated with the key reference does not change and the PIV Card Application returns '69 83'.

AS05.25A-R4: If the authentication data in the command data field does not match the current value of the reference data or if either the authentication data or the new reference data in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, the PIV Card Application shall not change the reference data associated with the key reference and shall return either status word '6A 80' or '63 CX', with the following restrictions:

(a) If the authentication data in the command data field satisfies the criteria in Section 2.4.3 of Part 2 of SP 800-73-4 and matches the current value of the reference data, but the new reference data in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4 the PIV Card Application shall return status word '6A 80'.

(b) If the authentication data in the command data field does not match the current value of the reference data, but both the authentication data and the new reference data in the command data field of the command satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, the PIV Card Application shall return status word '63 CX'.

(c) If status word '6A 80' is returned, the security status and retry counter associated with the key reference shall remain unchanged.

Required Vendor Information

VE05.25A-R4.01: The vendor shall state in its documentation that the card supports the assertions made in AS05.25A-R4.

Required Test Procedures

TE05.25A-R4.01: The tester shall validate that: 1) the vendor documentation contains the information required in VE05.25A-R4.01; and 2) the card returns status word '6A 80' or '63 CX' based on the conditions mentioned in AS05.25A-R4.

AS05.26: 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.

Required Vendor Information

VE05.26.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.26.

Required Test Procedures

TE05.26.01: The tester shall validate that the vendor documentation states the required information in VE05.26.01 and the retry counter associated with the key reference is set to the reset retry value associated with the key reference when the command succeeds.

AS05.27: If status word '63 CX' is returned, 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.

Required Vendor Information

VE05.27.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.27.

Required Test Procedures

TE05.27.01: The tester shall validate that the vendor's documentation contains the information required in VE05.27.01 and the retry counter associated with the key reference is decremented by one if the card command fails.

AS05.28: Moved to AS05.25A-R4.

AS05.28A: If the key reference is set to a value other than what is supported by the card, the PIV Card Application shall return the status word '6A 88'.

Required Vendor Information

VE05.28A.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.28A.

Required Test Procedures

TE05.28A.01: The tester shall validate that the vendor's documentation contains the information required in VE05.28A.01.

A.5.2.3 RESET RETRY COUNTER Card Command

AS05.29: The only key reference allowed in the P2 parameter of the RESET RETRY COUNTER command is the PIV Card Application PIN. The PIV Card Application may allow the reference data associated with other key references to be changed by the PIV Card Application RESET RETRY COUNTER, if PIV Card Application will only perform the command with other key references if the requirements specified in Section 2.9.2 of FIPS 201-2 are satisfied. If a key reference is specified in P2 that is not supported by the card, the PIV Card Application shall return the status word '6A 88'.

Required Vendor Information

VE05.29.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.29.

Required Test Procedures

TE05.29.01: The tester shall review the vendor's documentation and validate that includes the information required in VE05.29.01.

AS05.30: 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'.

Required Vendor Information

VE05.30.01: This information is requested as part of VE05.17.01.

VE05.30.02: The vendor shall specify in its documentation that the RESET RETRY COUNTER card command will not reset the PIN's retry counter and the card will return '69 83' (Reset operation blocked) when the PUK's retry counter is zero.

Required Test Procedures

TE05.30.01: The tester shall review the vendor's documentation and validate that the information requested in VE05.30.02 and VE05.30.01 are present. (NOTE: Testing this condition will leave the card unusable for further tests of the RESET RETRY COUNTER command since the reset counter is zero).

AS05.31: 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.

Required Vendor Information

VE05.31.01: This information is requested as part of VE05.17.01.

VE05.31.02: The vendor shall specify in its documentation that the card supports the assertion made in AS05.31.

Required Test Procedures

TE05.31.01: The tester shall validate that when the card command succeeds, the PIN's retry counter is set to the PIN's reset retry value specified in VE05.31.01, and the security status of the PIN's key reference is not changed. If the PUK's retry counter can be reset, the tester shall validate that the PUK's retry counter was reset to its initial reset retry value.

AS05.32: If the PIV Card Application returns status word '63 CX' then the retry counter associated with the PIN shall not be reset, 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.

Required Vendor Information

VE05.32.01: The vendor shall state in its documentation that card supports the assertion made in AS05.32.

Required Test Procedures

TE05.32.01: The tester shall validate that if the PIV Card Application returns '63 CX', then the retry counter associated with the PIN is not reset, the security status of the PIN's key reference is set to FALSE, and the PUK's retry counter is decremented by one.

AS05.33: If the reset retry counter authentication data (PUK) in the command data field of the command does not match reference data associated with the PUK, then the PIV Card Application shall return the status word '63 CX'. If the new reference data (PIN) in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the PIV Card Application shall return the status word '6A 80'. If the reset retry counter authentication data (PUK) in the command data field of the command does not match reference data associated with the PUK and the new reference data (PIN) in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the PIV Card Application shall return either status word '6A 80' or '63 CX'. If the PIV Card Application returns status word '6A 80', then the retry counter associated with the PIN shall not be reset, the security status of the PIN's key reference shall remain unchanged, and the PUK's retry counter shall remain unchanged.

Required Vendor Information

VE05.33.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.33.

Required Test Procedures

TE05.33.01: The tester shall review and validate that the vendor's documentation includes the information required in VE05.33.01 and the tester shall also validate that card inplements the RESET RETRY COUNTER card command in a manner consistent with AS05.33 by ensuring the following conditions:

(a) If the new reference data (PIN) in the command data field of the command satisfies the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, but the reset retry counter authentication data (PUK) in the command data field of the command does not match reference data associated with the PUK, then the PIV Card Application returns the status word '63 CX'.
(b) If the reset retry counter authentication data (PUK) in the command data field of the command matches reference data associated with the PUK, but the new reference data (PIN) in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the PIV Card Application returns the status word '6A 80'.
(c) If the reset retry counter authentication data (PUK) in the command data field of the command does not match reference data associated with the PUK and the new reference data (PIN) in the command data field of the command does not satisfy the criteria in Section 2.4.3 of Part 2 of SP 800-73-4, then the PIV Card Application returns either status word '6A 80' or '63 CX'.

(d) If the PIV Card Application returns status word '6A 80', then the retry counter associated with the PIN is not reset, the security status of the PIN's key reference remains unchanged, and the PUK's retry counter remains unchanged.

AS05.33A: Moved requirement into AS05.29.

A.5.2.4 GENERAL AUTHENTICATE Card Command

AS05.34: 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.

1) **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).**
2) **The GENERAL AUTHENTICATE command shall be used with the 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.**
3) **The GENERAL AUTHENTICATE command shall be used with the key management key ('9D') and the retired key management keys ('82' – '95') to realize key establishment primitives specified in SP 800-78-4 (ECDH and RSA).**
4) **The GENERAL AUTHENTICATE command shall be used with the PIV Secure Messaging key ('04') and cryptographic algorithm identifier '27' or '2E' to establish session keys for secure messaging as specified in Section 4 of SP 800-73-4, Part 2. If key reference '04' is specified in P2 then algorithm identifiers in P1 other than '27' and '2E' shall not be permitted and the PIV Card Application shall return the status word '6A 86'.**

Required Vendor Information

VE05.34.01: The vendor shall specify in its documentation the types of cryptographic operations (authentication, key establishment primitives, signing primitives, and secure messaging) supported by the card.

Required Test Procedures

TE05.34.01: The tester shall validate that the GENERAL AUTHENTICATE command is implemented to authenticate the card to the client application (Pertains to AS05.34-(1)).

TE05.34.02: The tester shall validate that the GENERAL AUTHENTICATE command is implemented to authenticate the client application to the card (Pertains to AS05.34-(1)).

TE05.34.03: The tester shall validate that the GENERAL AUTHENTICATE command is implemented to mutually authenticate the card to the client application and the client application to the card (Pertains to AS05.34-(1)).

TE05.34.04: The tester shall validate that the GENERAL AUTHENTICATE command is implemented to realize signing functionality (Pertains to AS05.34-(2)).

TE05.34.05: The tester shall validate that the GENERAL AUTHENTICATE command is implemented to support the RSA key transport or Elliptic Curve Diffie-Hellman key agreement primitives specified in SP 800-78-4 (Pertains to AS05.34-(3)).

TE05.34.06: If the '04' key is implemented, the tester shall validate that the GENERAL AUTHENTICATE command is implemented to support only cryptographic algorithm identifiers '27' and/or '2E' to establish session keys for secure messaging. The tester shall validate that if key reference '04' is specified in P2 and an algorithm identifier other than '27' or '2E' is specified in P1 then the card returns the status word '6A 86' (Pertains to AS05.34-(4)).

AS05.35: The GENERAL AUTHENTICATE command shall be implemented to realize the signing functionality on the PIV client application programming interface.

Note: This assertion is tested as part of AS05.34.

AS05.36: If an invalid value of algorithm reference (P1) and/or key reference (P2) is sent to the card, the PIV Card Application shall return the status word '6A 86'.

Required Vendor Information

VE05.36.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.36.

Required Test Procedures

TE05.36.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.36.01 and the card returns status word '6A 86' when an invalid value of algorithm reference (P1) or key reference (P2) is sent to the card.

AS05.36A: If an invalid value is sent in the data field, the PIV Card Application shall return the status word '6A 80'.

Required Vendor Information

VE05.36A.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.36A.

Required Test Procedures

TE05.36A.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.36A.01 and the card returns status word '6A 80' when an invalid value in data field of the command is sent to the card.

AS05.36B: If the command is used to authenticate the card to the client application using a PIN-protected PIV key without prior PIN or OCC verification the PIV Card Application shall return the status word '69 82'.

Required Vendor Information

VE05.36B.01: The vendor shall state in its documentation that the card supports the assertion made in AS05.36B.

Required Test Procedures

TE05.36B.01: The tester shall review the vendor's documentation and validate that it contains the information required in VE05.36B.01 and the card returns status word '69 82' whenever the command is used to authenticate the card to the client application using a PIN-protected key without prior PIN verification.

AS05.36C: If a card command other than the GENERAL AUTHENTICATE 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.

Required Vendor Information
VE05.36C.01: The vendor shall specify in its documentation that the card supports the assertion made in AS05.36C.

Required Test Procedures

TE05.36C.01: The tester shall review the vendor's documentation and validate that it states that the PIV Card Application reverts back to the state it was in if a command other than GENERAL AUTHENTICATE is received before the termination of a GENERAL AUTHENTICATE chain.

A.5.3 PIV Card Application Card Commands for Credential Initialization and Administration

A.5.3.1 PUT DATA Card Command

AS05.37: The PUT DATA card command completely replaces the data content of a single data object in the PIV Card Application with new content.

Required Vendor Information

VE05.37.01: The vendor shall specify in its documentation the format, encoding, and the parameters of the PUT DATA command supported by the card.

Required Test Procedures

TE05.37.01: The tester shall validate that the card complies with the PUT DATA command as defined in SP 800-73-4, Part 2.

A.5.3.2 GENERATE ASYMMETRIC KEY PAIR Card Command

AS05.38: 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.

Required Vendor Information

VE05.38.01: The vendor shall specify in its documentation the cryptographic mechanism identifiers (specified in Table 5, Part 1 of SP 800-73-4) that have been implemented on the card.

Required Test Procedures

TE05.38.01: The tester shall validate that the card implements the algorithms associated with identifiers specified as part of VE05.38.01 requirement and that the public key returned is formatted based on data object tags specified in Table 11, Part 2 of SP 800-73-4.

AS05.39: The public key of the generated key pair is returned as the response to the command.

Note: This assertion is tested as part of AS05.38.

AS05.40: If there is reference data currently associated with the key reference, it is replaced in full by the generated data.

Required Test Procedures

TE05.40.01: The tester shall validate that the initial key pair is replaced in full by the generated data, by issuing a GENERAL AUTHENTICATE command following a GENERATE ASYMMETRIC KEY PAIR command and verifying the cryptographic results using the public key returned by the GENERATE ASYMMETRIC KEY PAIR command.

A.5.4 Secure Messaging (SM)

AS05.41-R4: When secure messaging is established, the PIV Card Application shall authenticate to the relying system and a set of symmetric session keys will be established.

Required Vendor Information

VE05.41-R4.01: The vendor shall specify in its documentation whether the card implements secure messaging.

Required Test Procedures

Note: This assertion is tested as a part of establishing the VCI interface.

AS05.42-R4: When implemented, SM for non-card-management operations shall only be established using the PIV Secure Messaging key specified in Table 4b of SP 800-73-4, Part 1, and the SM protocol in accordance with the specifications in SP 800-73-4 Section 4 of Part 2.

Required Vendor Information

VE05.42-R4.01: The vendor shall specify in its documentation that secure messaging is implemented in accordance with AS05.42-R4.

Required Test Procedures

TE05.42-R4.01: The tester shall review the vendor's documentation and validate that the SM for non-card-management operations shall only be established using the PIV Secure Messaging key.

AS05.43-R4: The SW protocol is the status byte of the overall secure messaging command and response processing. It indicates if the secure messaging was performed successfully. If the processing was successful, it shall be '90 00'; otherwise, it shall be as follows: '68 82' if secure messaging is not supported; '69 82' if the security status is not satisfied; '69 87' if the expected secure messaging data objects are missing; and '69 88' if the secure messaging data objects are incorrect. If the command processing was unsuccessful, the card shall return one of the above errors without performing further secure messaging.

Required Vendor Information

VE05.43-R4.01: The vendor shall specify in its documentation that the card conforms to the assertion stated in AS05.43-R4.

Required Test Procedures

TE05.43-R4.01: The tester shall review the vendor's documentation and validate that the PIV Card returns the applicable SW identified in AS05.43-R4 without performing further secure messaging if the command processing was unsuccessful.

AS05.44-R4: If the PIV Card supports secure messaging, the PIV Secure Messaging key shall be generated on the PIV Card and the PIV Card shall not permit exportation of the PIV Secure Messaging Key.

Required Vendor Information

VE05.44-R4.01: The vendor shall specify in its documentation that the card conforms to the assertion stated in AS05.44-R4.

Required Test Procedures

TE05.44-R4.01: The tester shall review the vendor's documentation and validate that the PIV Secure Messaging key is generated on the PIV Card and the PIV Card does not permit exportation of the PIV Secure Messaging Key.

AS05.45-R4: The cryptographic operations that use the PIV Secure Messaging key shall be available through the contact and contactless interfaces of the PIV Card.

Note: This assertion is not separately tested since it is tested as a part of initiating the SM and VCI interfaces.

Appendix B—PIV Client API Test Assertions

All tests in Appendices B.1 to B.10 are performed over the contact interface only except where stated otherwise. These tests apply to both PIV Middleware versions (with and without SM/VCI). Tests in Appendix B.11 are performed for PIV Middleware version "800-73-4 Client API with SM" only, using a contactless reader interface.

Test Assertion Template

Purpose	A quick description of the test and why it is being run.
Target	`The PIV client API function call being tested.`
Reference(s)	References to SP 800-73-4 or other relevant publications.
Precondition(s)	Anything that must be done or known prior to executing the scenario.
Test Steps	`Sequence of steps for making a function call.`
Expected Result(s)	What the expected execution path yields in terms of data (if applicable) and response status codes.
Postcondition(s)	A description of the PIV Middleware's client application and card application state once the test scenario completes.

B.1 pivConnect

B.1.1 Valid Path Test Assertions

B.1.1.1 Initiate Exclusive Connection

Purpose	Confirm that an exclusive connection can be obtained by a calling application to the PIV Card reader.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. A valid connection description is provided for the card reader. 2. There exists a valid physical connection between an instance of the PIV Card and the client application. 3. No application is currently connected to the PIV Card Application.

Test Steps	`1. Set sharedConnection := false`
	`2. Set connectionDescription := <<valid connection>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *`(IN)`* `sharedConnection`
	• *`(INOUT)`* `connectionDescription`
	• *`(INOUT) CDLength`*
	• *`(OUT)`* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_OK` and initialized `cardHandle`.
Postcondition(s)	Client application is connected to PIV Card.

B.1.1.2 Initiate Shared Connection

Purpose	Confirm that a shared connection can be established by two distinct client applications to the PIV Card with a specific ICC.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. A valid connection description is provided for the card reader. 2. There exists a valid physical connection between an instance of the PIV Card and a client application. 3. Another client application is currently connected via a shared connection to the PIV Card Application.

Test Steps	`1. Set sharedConnection := true`
	`2. Set connectionDescription := <<valid connection>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *`(IN)`* `sharedConnection`
	• *`(INOUT)`* `connectionDescription`
	• *`(INOUT) CDLength`*
	• *`(OUT)`* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_OK` and initialized `cardHandle`.
Postcondition(s)	Both client applications are connected through the same connection to the PIV Card Application.

B.1.2 Test Assertions for Error Conditions

B.1.2.1 Malformed Connection Description

Purpose	Confirm that the correct status word is returned when a malformed connection description is used.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. An invalid connection description is provided for the card reader. 2. There exists a valid physical connection between an instance of the PIV Card and the client application.

Test Steps	`1. Set sharedConnection := true | false`
	`2. Set connectionDescription := <<invalid connection>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *`(IN)`* `sharedConnection`
	• *`(INOUT)`* `connectionDescription`
	• *`(INOUT) CDLength`*
	• *`(OUT)`* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_CONNECTION_DESCRIPTION_MALFORMED`.
Postcondition(s)	1. The `cardHandle` variable is not initialized.

	2. The client application is not connected to the PIV Card Application.

B.1.2.2 Attempting to Share/Lock an Exclusive Connection

Purpose	Ensure that when an exclusive connection is initially established that no additional connections can be established.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. A valid connection description is provided for the card reader. 2. There exists a valid physical connection between an instance of the PIV Card and the client application. 3. An application owns an exclusive connection (`sharedConnection := false`).

Test Steps	`1. Set sharedConnection := true | false`
	`2. Set connectionDescription := <<valid connection>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *`(IN)`* `sharedConnection`
	• *`(INOUT)`* `connectionDescription`
	• *`(INOUT) CDLength`*
	• *`(OUT)`* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_CONNECTION_LOCKED`.
Postcondition(s)	1. The client application previously connected remains connected. 2. The `cardHandle` variable is not initialized. 3. The requesting client application is not connected to the PIV Card Application.

B.1.2.3 Attempting to Lock a Shared Connection

Purpose	Ensure that the PIV Middleware does not lock a PIV Card Application connection that has an open shared connection.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. A valid connection description is provided for the card reader. 2. There exists a valid physical connection between an instance of the PIV Card and the client application. 3. A client application owns a shared connection (`sharedConnection := true`).

Test Steps	`1. Set sharedConnection := false`
	`2. Set connectionDescription := <<valid connection>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *`(IN)`*`sharedConnection`

	• *(INOUT)* `connectionDescription` • *(INOUT) CDLength* • *(OUT)* `cardHandle`
Expected Result(s)	Call returns with status_word of `PIV_CONNECTION_FAILURE`.
Postcondition(s)	1. The client application previously connected remains connected. 2. The `cardHandle` variable is not initialized. 3. The requesting client application is not connected to the PIV Card Application.

B.1.2.4 Attempting to Open an Unsupported Connection

Purpose	Confirm that the PIV Middleware returns the correct status word when an unsupported connection mode is attempted.
Target	`pivConnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.2 2. AS04.01, AS04.02, AS04.02A-R4
Precondition(s)	1. An invalid connection mode (e.g., Integrated Services Digital Network (ISDN)) is attempted. 2. There exists a valid physical connection between an instance of the PIV Card and the client application.

Test Steps	`1. Set sharedConnection := true | false`
	`2. Set connectionDescription := <<valid ISDN connection string>>`
	`3. Create cardHandle reference`
	`4. Call pivConnect with`
	• *(IN)* `sharedConnection`
	• *(INOUT)* `connectionDescription`
	• *(INOUT) CDLength*
	• *(OUT)* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_CONNECTION_FAILURE`.
Postcondition(s)	1. The `cardHandle` variable is not initialized. 2. The client application is not connected to the PIV Card.

B.2 pivDisconnect

B.2.1 Valid Test Assertions

B.2.1.1 Disconnect an Exclusive Connection

Purpose	Ensure that the PIV Middleware closes a currently open exclusive PIV Card Application connection.
Target	`pivDisconnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.3 2. AS04.01, AS04.02A-R4, AS04.03

Precondition(s)	1. There exists a valid physical and exclusive connection between an instance of the PIV Card and the client application. 2. The client application currently has a connection accessible through `cardHandle`.
Test Steps	`1. Call pivDisconnect with arguments` • *`(IN)`* `cardHandle`
Expected Result(s)	Call returns with status_word of `PIV_OK`.
Postcondition(s)	The client application is no longer connected to the PIV Card Application.

B.2.1.2 Disconnect a Shared Connection

Purpose	Ensure that the PIV Middleware closes an open and shared PIV Card Application connection without impacting other client application's connections to that same PIV Card Application.
Target	`pivDisconnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.3 2. AS04.03
Precondition(s)	1. There exists a valid physical shared connection between an instance of the PIV Card and the client application. 2. At least two distinct client applications (having two distinct `cardHandle` references) are also connected to the PIV Card Application.
Test Steps	`1. Call pivDisconnect with arguments` • *`(IN)`* `cardHandle`
Expected Result(s)	Call returns with status_word of `PIV_OK`.
Postcondition(s)	1. The client application is no longer connected to the PIV Card Application. 2. All other client applications maintain their previously valid connections.

B.2.2 Test Assertions for Error Cases

B.2.2.1 Attempt Disconnect with Invalid Card Handle

Purpose	Ensure that the PIV Middleware detects an invalid `cardHandle` argument.
Target	`pivDisconnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.3 2. AS04.03
Precondition(s)	1. There exists a valid physical connection between an instance of the PIV Card and the client application. 2. A client application currently has a connection accessible through `cardHandle`.

Test Steps	`1. Set cardHandle := <<invalid cardHandle>>`
	`2. Call pivDisconnect with`
	• *(IN)* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The client application remains connected to the PIV Card Application.

B.2.2.2 Disconnecting a Disconnected Client Application

Purpose	Verify that when the client application tries to close a closed PIV Card Application connection (i.e., with the same `cardHandle`), the PIV Middleware returns an Invalid Card Handle message.
Target	`pivDisconnect`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.3 2. AS04.03
Precondition(s)	1. A client application with a valid and open `cardHandle` to a PIV Card Application that was previously closed. 2. The card is physically connected to the card reader.
Test Steps	`1. Call pivDisconnect with arguments` • *(IN)* `cardHandle`
Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The client application remains unconnected to the PIV Card Application.

B.3 pivSelectCardApplication

B.3.1 Valid Test Assertions

B.3.1.1 Select a Card Application with a Full AID

Purpose	Ensure that the PIV Middleware locates and selects a valid PIV Card Application, stores its properties, and returns a reference to the application properties.
Target	`pivSelectCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.1 2. AS04.01, AS04.02A-R4, AS04.04
Precondition(s)	1. The client application owns a connection accessible through `cardHandle` through a contact reader.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set applicationID := <<AID of PIV Card Application>>`
	`3. Create applicationProperties reference`
	`4. Call pivSelectCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `applicationAID`
	• *(IN)* *`aidLength`*
	• *(OUT)* `applicationProperties`

	• *(INOUT) APLength*
Expected Result(s)	Call returns with status_word of `PIV_OK` and initialized `applicationProperties` reference.
Postcondition(s)	The "currently selected application" of the PIV Card is the PIV Card Application. The PIV Card Application's security state is established.

B.3.1.2 Use a Right Truncated AID to Select a Card Application

Purpose	Ensure that the PIV Middleware is able to locate and select a valid PIV Card Application that is identified by a right truncated AID, store its properties, and return a reference via the `applicationProperties` function parameter.
Target	`pivSelectCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.1 2. AS04.04
Precondition(s)	1. The client application owns a connection accessible through `cardHandle`.
Test Steps	1. Set cardHandle := <<valid cardHandle>> 2. Set applicationID := <<right truncated AID of PIV Card Application>> 3. Create applicationProperties reference 4. Call pivSelectCardApplication with • *(IN)* cardHandle • *(IN)* applicationAID • *(IN) aidLength* • *(OUT)* applicationProperties • *(INOUT) APLength*
Expected Result(s)	Call returns with status_word of `PIV_OK` and sets the `applicationProperties` reference.
Postcondition(s)	The "currently selected application" of the PIV Card is the PIV Card Application. The PIV Card Application's security state is established.

B.3.2 Test Assertions for Error Conditions

B.3.2.1 Detect and Handle an Invalid cardHandle Reference

Purpose	Ensure that the PIV Middleware detects and gracefully exits when passed an invalid `cardHandle`.
Target	`pivSelectCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.1 2. AS04.04
Precondition(s)	1. There exists a valid physical connection between an instance of the PIV Card and the client application. 2. A client application currently has a connection accessible through `cardHandle`.

Test Steps	`1. Set cardHandle := <<invalid cardHandle>>`
	`2. Set applicationID := <<AID of PIV Card Application>>`
	`3. Create applicationProperties reference`
	`4. Call pivSelectCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `applicationAID`
	• *(IN) aidLength*
	• *(OUT)* `applicationProperties`
	• *(INOUT) APLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE` and does not initialize `applicationProperties` reference.
Postcondition(s)	The client application remains in the state it had prior to calling `pivSelectCardApplication`.

B.3.2.2 Detect and Handle an Invalid applicationAID

Purpose	Ensure that the PIV Middleware detects and gracefully exits when passed an invalid `applicationAID`.
Target	`pivSelectCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.1 2. AS04.04
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set applicationID := <<invalid applicationID>>`
	`3. Create applicationProperties reference`
	`4. Call pivSelectCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `applicationAID`
	• *(IN) aidLength*
	• *(OUT)* `applicationProperties`
	• *(INOUT) APLength*

Expected Result(s)	Call returns with status_word of `PIV_CARD_APPLICATION_NOT_FOUND` and does not set the `applicationProperties` reference.
Postcondition(s)	The client application remains in the state it had prior to calling `pivSelectCardApplication.`

B.3.2.3 Identify and Handle an Insufficient Buffer

Purpose	Ensure that the PIV Middleware identifies and handles an insufficient allocated buffer for the application property template.
Target	`pivSelectCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.1 2. AS04.04A-R4
Precondition(s)	1. The client application owns a connection accessible through `cardHandle`.

	2. Length of the buffer allocated for data by the client application is only 1 byte.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set applicationID := <<AID of PIV Card Application>>`
	`3. Create applicationProperties reference`
	`4. Call pivSelectCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `applicationAID`
	• *(IN) `aidLength`*
	• *(OUT)* `applicationProperties`
	• *(INOUT) `APLength`*

Expected Result(s)	Call returns with status word of `PIV_INSUFFICIENT_BUFFER` and sets the value of the *`APLength`* parameter to the length of the application properties.
Postcondition(s)	The PIV Card Application is selected.

B.4 pivLogIntoCardApplication

B.4.1 Valid Test Assertions

B.4.1.1 Log on to the Card Application

Purpose	Validate that the PIV Middleware initiates updates to the security status(es) with the PIV Card Application.
Target	`pivLogIntoCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.3 2. AS03.03, AS04.01, AS04.02A-R4, AS04.05
Precondition(s)	1. The card has established a connection to the client. 2. The `cardHandle` was properly initialized by `pivConnect`. 3. The client application has successfully executed the `pivSelectCardApplication` command.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Set authenticators := <<valid authenticators byte sequence for PIV Card Application PIN>>`
	`3. Call pivLogIntoCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `authenticators`
	• *(IN) `AuthLength`*
	`4. Logout and reset security conditions for PIV Middleware and PIV Card Application`
	`5. Repeat steps 1 through 4 using all remaining valid authenticators (Global PIN, pairing code, OCC data)`

Expected Result(s)	Call returns with status_word of `PIV_OK`.
Postcondition(s)	Security context is updated and the client application can now perform read operations on PIN-protected data objects controlled by the PIV Card Application. The client is thus logged into the PIV Card Application.

	Note: Use of the pairing code with pivLogIntoCardApplication does not enable read access to PIN-protected data objects. Also, biometric data objects will not be accessible when using OCC data as an authenticator.

B.4.2 Test Assertions for Error Conditions

B.4.2.1 Attempt Logon with an Invalid cardHandle

Purpose	Ensure the PIV Middleware detects and processes an invalid card handle.
Target	`pivLogIntoCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.3 2. AS04.05
Precondition(s)	1. The card has established a connection to the client. 2. The `cardHandle` was properly initialized by `pivConnect`. 3. The client application has successfully executed the `pivSelectCardApplication` command.

Test Steps	`1. Set cardHandle := <<an invalid cardHandle>>`
	`2. Set authenticators := <<valid authenticators byte sequence>>`
	`3. Call pivLogIntoCardApplication with`
	• *(IN)* `cardHandle`
	• *(IN)* `authenticators`
	• *(IN) AuthLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The client application is not logged into the PIV Card Application and was not able to update the application security status with the PIV Card Application.

B.4.2.2 Attempt Logon with a Malformed Authenticator

Purpose	Ensure the PIV Middleware detects and processes a malformed authenticator byte sequence.
Target	`pivLogIntoCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.3 2. AS04.05
Precondition(s)	1. The card has established a connection to the client. 2. The `cardHandle` was properly initialized by `pivConnect`. 3. The client application has successfully executed the `pivSelectCardApplication` command.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Set authenticators := <<a malformed authenticators byte sequence>>`
	`3. Call pivLogIntoCardApplication with`

	• *(IN)* `cardHandle` • *(IN)* `authenticators` • *(IN) `AuthLength`*
Expected Result(s)	Call returns with status_word of `PIV_AUTHENTICATOR_MALFORMED`.
Postcondition(s)	The client application is not logged into the PIV Card Application and was not able to update the application security status of the PIV Card Application.

B.4.2.3 Attempt Logon with Invalid Authenticator

Purpose	Ensure PIV Middleware detects and processes an authenticator that has the correct format but does not result in a valid security permission/context.
Target	`pivLogIntoCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.3 2. AS04.05
Precondition(s)	1. The card has established a connection to the client. 2. The `cardHandle` was properly initialized by `pivConnect`. 3. The client application has successfully executed the `pivSelectCardApplication` command.
Test Steps	1. `Set cardHandle := <<a valid cardHandle>>` 2. `Set authenticators := <<a well formed authenticators byte sequence containing an invalid PIN and/or Key Reference value>>` 3. `Call pivLogIntoCardApplication with` • *(IN)* `cardHandle` • *(IN)* `authenticators` • *(IN) `AuthLength`*
Expected Result(s)	Call returns with status_word of `PIV_AUTHENTICATION_FAILURE`.
Postcondition(s)	The client application is not logged into the PIV Card Application and was not able to update the application security status of the PIV Card Application.

B.5 pivLogoutOfCardApplication

B.5.1 Valid Test Assertions

B.5.1.1 Log out of the Card Application

Purpose	Reset security context of the PIV Card Application.
Target	`pivLogoutOfCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.5 2. AS04.01, AS04.02A-R4, AS04.07
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the card application.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Call pivLogoutOfCardApplication with`
	• *(IN)* `cardHandle`
	`3. Set OID := <<valid OID for each of the following objects: Fingerprints, Facial Image, Printed Information, Iris Images, Pairing Code Reference Data Container>>`
	`4. Create data reference`
	`5. Call pivGetData with (each data object identified in step 3)`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(OUT)* `data`
	• *(INOUT) DataLength*

Expected Result(s)	Step 2: Call returns with status_word of `PIV_OK` and the client application is logged off of the PIV Card Application. Step 5: Call returns with status_word:= `PIV_SECURITY_CONDITION_NOT_SATISFIED` and does not set data reference.
Postcondition(s)	1. The `cardHandle` remains valid. 2. The connection remains open.

B.5.1.2 Attempt Log Out Without Logging In

Purpose	Verify that the PIV Middleware does not return an error when client application requests a logout without first logging in.
Target	`pivLogoutOfCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.5 2. AS04.07
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client has successfully executed the `pivSelectCardApplication` command. 3. The client is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Call pivLogoutOfCardApplication with`
	• *(IN) cardHandle*

Expected Result(s)	Call returns with status_word of `PIV_OK`.
Postcondition(s)	The precondition states remain unchanged (only "free read" data can be read).

B.5.2 Test Assertions for Error Conditions

B.5.2.1 Attempt Log Out with Invalid cardHandle

Purpose	Ensure the PIV Middleware detects and handles an invalid `cardHandle`.

Target	`pivLogoutOfCardApplication`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.5 2. AS04.07
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the card application. 3. The client has established an "application security status."

Test Steps	`1. Set cardHandle := <<an invalid cardHandle>>`
	`2. Call pivLogoutOfCardApplication with`
	• *`(IN)`* `cardHandle`

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The precondition states remain unchanged.

B.6 pivGetData

B.6.1 Valid Test Assertions

B.6.1.1 Get a Reference to Data Object that Does Not Require Login

Purpose	Ensure the PIV Middleware reads data objects from the PIV Card Application that do not require a login.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.01, AS04.02A-R4, AS04.06
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. The client is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<valid OID>> (Repeat this for all implemented objects on the card except for Fingerprints, Printed Information, Facial Image, Iris Images, and Pairing Code Reference Data Container)`
	`3. Create data reference`
	`4. Call pivGetData with (each data object identified in Step 2)`
	• *`(IN)`* `cardHandle`
	• *`(IN)`* `OID`
	• *`(IN) oidLength`*
	• *`(OUT)`* `data`
	• *`(INOUT) DataLength`*

Expected Result(s)	Call returns with status_word of `PIV_OK` in each case and sets reference to data.
Postcondition(s)	N/A

B.6.1.2 Get a Reference to Data Object that Requires Login

Purpose	Ensure the PIV Middleware reads data objects from the card that require a login.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS02.03, AS04.01, AS04.02A-R4, AS04.06
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. The client is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set authenticators := <<valid authenticators byte sequence for PIV Card Application PIN>>`
	`3. Call pivLogIntoCardApplication with`
	• `(IN) cardHandle`
	• *`(IN) authenticators`*
	• `(IN) AuthLength`
	`4. Set OID := <<valid OID>> (Repeat this for all implemented objects in the following set - Fingerprints, Printed Information, Facial Image, Iris Images, and Pairing Code Reference Data Container)`
	`5. Create data reference`
	`6. Call pivGetData with (each data object identified in step 4)`
	• *`(IN)`* `cardHandle`
	• *`(IN)`* `OID`
	• *`(IN) oidLength`*
	• *`(OUT)`* `data`
	• *`(INOUT) DataLength`*
	`7. Logout and reset security conditions for PIV Middleware and PIV Card Application`
	`8. Repeat steps 2 through 7 using the Global PIN`
	`9. Repeat steps 2 through 7 using the OCC data`

Expected Result(s)	Step 6: Call returns with status_word of `PIV_OK` in all cases and sets reference to data. Step 8: Call returns with status_word of `PIV_OK` in all cases and sets reference to data. Step 9: • Call returns with status_word of `PIV_OK` in response to the requests for the Printed Information and Pairing Code Reference Data Container and sets reference to data. • Call returns with status_word of `PIV_SECURITY_CONDITIONS_NOT_SATISFIED` in response to the requests for the Cardholder Fingerprints, Cardholder Facial

	Image, and Cardholder Iris Images and does not set the data reference.
Postcondition(s)	The client application is logged off of the PIV Card Application. Only "free read" data can be read.

B.6.2 Test Assertions for Error Conditions

B.6.2.1 Identify and Handle an Invalid cardHandle

Purpose	Ensure the PIV Middleware recognizes and handles an invalid `cardHandle`.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. The client application is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<invalid cardHandle>>`
	`2. Set OID := <<valid OID>>`
	`3. Create data reference`
	`4. Call pivGetData with`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(OUT)* `data`
	• *(INOUT) DataLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE` and does not initialize data reference.
Postcondition(s)	The precondition states remain unchanged.

B.6.2.2 Identify and Handle an Invalid Object Identifier

Purpose	Ensure the PIV Middleware recognizes and handles an invalid OID.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application is logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<invalid OID>> (Improper syntax or not found in Table 3 of SP 800-73-4 Part 1)`

	3. Create data reference 4. Call pivGetData with • *(IN)* cardHandle • *(IN)* OID • *(IN) oidLength* • *(OUT)* data • *(INOUT) DataLength*
Expected Result(s)	Call returns with status_word of `PIV_INVALID_OID` and does not set data reference.
Postcondition(s)	The client application remains in the state it had before the call.

B.6.2.3 The Client Application Handles Missing Data Object

Purpose	Ensure the PIV Middleware recognizes and handles a missing OID.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06
Precondition(s)	1. The PIV Card does not have a container for one (or more) optional data object. 2. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 3. The client is logged into the PIV Card Application.
Test Steps	1. Set cardHandle := <<valid cardHandle>> 2. Set OID := <<valid OID>> (Found in Table 3 of SP 800-73-4 Part 1 that is not present on the PIV Card (i.e., no container is allocated) 3. Create data reference 4. Call pivGetData with • *(IN)* cardHandle • *(IN)* OID • *(IN) oidLength* • *(OUT)* data • *(INOUT) DataLength*
Expected Result(s)	Call returns with status_word of `PIV_DATA_OBJECT_NOT_FOUND` and does not initialize data reference.
Postcondition(s)	The client application remains in the state it had before the call.

B.6.2.4 The Client Application Handles Zero-Length Data Object

Purpose	Ensure the PIV Middleware recognizes and handles a data object that has a container, but is not used.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06
Precondition(s)	1. The PIV Card has containers for one or more optional data objects, but the data objects have not been used.

	2. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 3. The client is logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<valid OID>> (Found in Table 3 of SP 800-73-4 Part 1 that is present but not used on the PIV Card`
	`3. Create data reference`
	`4. Call pivGetData with`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(OUT)* `data`
	• *(INOUT) DataLength*

Expected Result(s)	Call returns with status_word of `PIV_DATA_OBJECT_NOT_FOUND` and does not initialize data reference.
Postcondition(s)	The client application remains in the state it had before the call.

B.6.2.5 Security Conditions are Enforced for Secured Objects

Purpose	Ensure that security conditions are enforced for retrieving data from secured applications.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06
Precondition(s)	1. The client application currently owns a connection to the PIV Card Application accessible through `cardHandle.` 2. The client application has successfully selected the PIV Card Application. 3. The client is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<valid OID for each of the following objects: Fingerprints, Facial Image, Printed Information, Iris Images, and the Pairing Code Reference Data Container>>`
	`3. Create data reference`
	`4. Call pivGetData with (each data object identified in step 2)`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(OUT)* `data`
	• *(INOUT) DataLength*

Expected Result(s)	Call returns with status_word of `PIV_SECURITY_CONDITION_NOT_SATISFIED` and does not set data reference.
Postcondition(s)	The client application remains in the state it had before the call.

B.6.2.6 Identify and Handle an Insufficient Buffer

Purpose	Ensure that the PIV Middleware identifies and handles an insufficient buffer for data retrieved from the card.
Target	`pivGetData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.4 2. AS04.06A-R4
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Length of the buffer allocated for data by the client application is only 1 byte.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<OID of X.509 Certificate for Card Authentication>>`
	`3. Create data reference`
	`4. Call pivGetData with`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(OUT)* `data`
	• *(INOUT) DataLength*

Expected Result(s)	Call returns with status_word of `PIV_INSUFFICIENT_BUFFER` and does not initialize data reference, but sets the `DataLength` parameter to the length of the retrieved data.
Postcondition(s)	The precondition states are unaffected.

B.7 pivPutData

B.7.1 Valid Test Assertions

B.7.1.1 Write Data to an Object on the Card through the Client Application

Purpose	Ensure the PIV Middleware writes the entire data content to an object on the PIV Card Application.
Target	`pivPutData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.1 2. AS04.01, AS04.02A-R4, AS04.09
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.
Test Steps	`1. Set cardHandle := <<valid cardHandle>>`

	2. Set OID := <<valid OID>> 3. Set data := <<a correctly formatted byte sequence> 4. Call pivPutData with (for all data objects) • *(IN)* cardHandle • *(IN)* OID • *(IN) oidLength* • *(IN)* data • *(IN) dataLength*
Expected Result(s)	Call returns with status_word of `PIV_OK` for each test case.
Postcondition(s)	Validate that the PIV Card Application has written the entire dataset of the selected object requested by the client application by issuing `pivGetData` function call.

B.7.2 Test Assertions for Error Conditions

B.7.2.1 Identify and Handle an Invalid cardHandle

Purpose	Ensure the PIV Middleware identifies and responds to an invalid card handle.
Target	`pivPutData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.1 2. AS04.09
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.
Test Steps	1. Set cardHandle := <<invalid cardHandle>> 2. Set OID := <<valid OID>> 3. Set data := <<a correctly formatted byte sequence> 4. Call pivPutData with • *(IN)* cardHandle • *(IN)* OID • *(IN) oidLength* • *(IN)* data • *(IN) dataLength*
Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The precondition states remain unchanged.

B.7.2.2 Identify and Handle an Invalid Object Identifier (OID)

Purpose	Ensure the PIV Middleware identifies and handles an invalid OID.
Target	`pivPutData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.1 2. AS04.09

Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<invalid OID>> (Improper syntax or not found in Table 3 of SP 800-73-4 Part 1)`
	`3. Set data := <<a correctly formatted byte sequence>`
	`4. Call pivPutData with`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(IN)* `data`
	• *(IN) dataLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_OID`.
Postcondition(s)	1. The PIV Card Application remains in the state it had prior to the `pivPutData` function call. 2. The precondition states remain unchanged.

B.7.2.3 Security Conditions are Enforced for Writing Data to the On-card Data Containers

Purpose	Ensure that security conditions are enforced for writing data to the PIV Card Application.
Target	`pivPutData`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.1 2. AS04.09
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. The PIV Card Application has not authenticated the PIV Card Application Administrator.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set OID := <<valid OID>>`
	`3. Create data reference`
	`4. Call pivPutData with (for all data objects)`
	• *(IN)* `cardHandle`
	• *(IN)* `OID`
	• *(IN) oidLength*
	• *(IN)* `data`
	• *(IN) dataLength*

Expected Result(s)	All calls return with status_word := `PIV_SECURITY_STATUS_NOT_SATISFIED` and does not initialize data reference.
Postcondition(s)	The client application remains in the state it had before the call.

B.8 pivGenerateKeyPair

B.8.1 Valid Test Assertions

B.8.1.1 Generate an Asymmetric Key Pair

Purpose	Ensure the PIV Middleware initiates generation of an asymmetric key pair on the PIV Card.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2 2. AS04.01, AS04.02A-R4, AS04.10
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<'9A'>>`
	`3. Set cryptographicMechanism:= <<'07' or '11'>>`
	`4. Create publicKey reference`
	`5. Call pivGenerateKeyPair with`
	• *(IN)* `cardHandle`
	• *(IN)* `keyReference`
	• *(IN)* `cryptographicMechanism`
	• *(OUT)* `publicKey`
	• *(INOUT) KeyLength*
	6. *Repeat steps 1 through 5 for key references '04', '9C', '9D', and '9E', using an appropriate cryptographic mechanism identifier from Table 5 in SP 800-73-4 Part 1 for the key reference.*

Expected Result(s)	Each call returns with status_word of `PIV_OK` and a reference to the publicKey.
Postcondition(s)	A public key / private key pair is created on the card and the private key is accessible to the client application with the applicable reference.

B.8.2 Test Assertions for Error Conditions

B.8.2.1 Identify and Handle an Invalid cardHandle

Purpose	Ensure the PIV Middleware catches invalid card handles.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2 2. AS04.10
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place

Test Steps	`1. Set cardHandle := <<invalid cardHandle>>`
	`2. Set keyReference := <<an existing key reference suitable for use with the specified cryptographicMechanism >>`
	`3. Set cryptographicMechanism := <<a recognized cryptographic mechanism identifier>>`
	`4. Create publicKey reference`
	`5. Call pivGenerateKeyPair with`
	• *(IN)* `cardHandle`
	• *(IN)* `keyReference`
	• *(IN)* `cryptographicMechanism`
	• *(OUT)* `publicKey`
	• *(INOUT) KeyLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	The precondition states are unaffected.

B.8.2.2 Identify and Handle an Invalid keyReference or Algorithm Combination

Purpose	Ensure that the PIV Middleware identifies and handles an invalid key reference.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2 2. AS04.10
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<a key reference not found in the specification>>`
	`3. Set cryptographicMechanism := <<a recognized cryptographic mechanism identifier>>`
	`4. Create publicKey reference`
	`5. Call pivGenerateKeyPair with`
	• *(IN)* `cardHandle`
	• *(IN)* `keyReference`

	• *(IN)* `cryptographicMechanism` • *(OUT)* `publicKey` • *(INOUT) `KeyLength`*
Expected Result(s)	Call returns with status_word of `PIV_INVALID_KEY_OR_KEYALG_COMBINATION`.
Postcondition(s)	1. The PIV Card Application remains in the state it had prior to the `pivGenerateKeyPair` function call. 2. The precondition states are unaffected.

B.8.2.3 Identify and Handle an Invalid cryptographicMechanism

Purpose	Ensure that the PIV Middleware identifies and handles unsupported cryptographic mechanism identifiers.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2 2. AS04.10
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<a valid key reference>>`
	`3. Set cryptographicMechanism := <<an unrecognized cryptographic mechanism identifier>>`
	`4. Create publicKey reference`
	`5. Call pivGenerateKeyPair with`
	• *(IN)* `cardHandle`
	• *(IN)* `keyReference`
	• *(IN)* `cryptographicMechanism`
	• *(OUT)* `publicKey`
	• *(INOUT) `KeyLength`*

Expected Result(s)	Call returns with status_word of `PIV_UNSUPPORTED_CRYPTOGRAPHIC_MECHANISM`.
Postcondition(s)	1. The PIV Card Application remains in the state it had prior to the `pivGenerateKeyPair` function call. 2. The precondition states are unaffected.

B.8.2.4 Security Conditions are Enforced

Purpose	Ensure that the PIV Middleware enforces the necessary security conditions when called from client application.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2

	2. AS04.10
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. The PIV Card Application has not authenticated the PIV Card Application Administrator.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set cryptographicMechanism := <<a recognized cryptographic mechanism identifier>>`
	`3. Set keyReference := <<a reference to a valid key that is associated with the selected cryptographicMechanism >>`
	`4. Create publicKey reference`
	`5. Call pivGenerateKeyPair with`
	• *(IN)* `cardHandle`
	• *(IN)* `keyReference`
	• *(IN)* `cryptographicMechanism`
	• *(OUT)* `publicKey`
	• *(INOUT) KeyLength*

Expected Result(s)	Call returns with status_word of `PIV_SECURITY_CONDITIONS_NOT_SATISFIED.`
Postcondition(s)	1. The Card Application remains in the state it had prior to the `pivGenerateKeyPair` function call. 2. The precondition states are unaffected.

B.8.2.5 Identify and Handle an Insufficient Buffer

Purpose	Ensure that the PIV Middleware identifies and handles an insufficient buffer.
Target	`pivGenerateKeyPair`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.4.2 2. AS04.10A-R4
Precondition(s)	1. The client application owns a connection to the Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Mutual authentication with the client application using the PIV Card Application Administration key has taken place. 4. Length of the buffer allocated for data by the client application is only 1 byte.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<an existing key reference suitable for use with the specified cryptographicMechanism>>`
	`3. Set cryptographicMechanism := <<a recognized cryptographic mechanism identifier>>`
	`4. Set KeyLength := <<1>>`
	`5. Create publicKey reference`

	6. `Call pivGenerateKeyPair with` • *(IN)* `cardHandle` • *(IN)* `keyReference` • *(IN)* `cryptographicMechanism` • *(OUT)* `publicKey` • *(INOUT) KeyLength*
Expected Result(s)	Call returns with status_word of `PIV_INSUFFICIENT_BUFFER` and sets the `KeyLength` parameter to the length of the returned public key.
Postcondition(s)	1. A new key pair is generated. 2. The precondition states are unaffected.

B.9 pivCrypt

B.9.1 Valid Test Assertions

B.9.1.1 Authenticate the Card Application to Client Application

Purpose	Exercise the PIV Middleware to perform Internal Authenticate.
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.01, AS04.02A-R4, AS04.08
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the PIV Card Application (required for step 1 for the '9A' PIV Authentication key).
Test Steps	1. `Set cardHandle := <<valid cardHandle>>` 2. `Set keyReference := <<'9A'>>` 3. `Set algorithmIdentifier := <<'07' or '11'>>` 4. `Set algorithmInput := <<Use the Dynamic Authentication Template format (Table 7 of SP 800-73-4 Part 2) to encode a challenge to be sent to the card>>` 5. `Create algorithmOutput reference` 6. `Call pivCrypt with` • *(IN)* `cardHandle` • *(IN)* `algorithmIdentifier` • *(IN)* `keyReference` • *(IN)* `algorithmInput` • *(IN) inputLength* • *(OUT)* `algorithmOutput` • *(INOUT) outputLength* 7. `Repeat steps 1 – 6, but with the '9E' key (Card Authentication key), and algorithmIdentifier := <<'00', '03', '07', '08', '0A', '0C', or '11'>>` 8. `Perform pivLogIntoCardApplication with PIV Card Application PIN and repeat steps 1 – 6, but with the '9C' key (digital signature key), algorithm <<'07', '11' or '14'>> and data to sign instead of a challenge` 9. `Repeat steps 1 – 6, but with the '9D' key (key management key), algorithm <<'07', '11' or '14'>> and an encrypted`

	key (with algorithm '07') or a public key (with algorithms '11' or '14') instead of a challenge
Expected Result(s)	Call returns with status_word of `PIV_OK` with the `algorithmOutput` carrying the signed challenge, transported key, shared secret Z, or a signature from the card.
Postcondition(s)	N/A

B.9.1.2 Mutual Authentication of Client Application and Card Application

Purpose	Exercise the PIV Card Application to perform Mutual Authenticate.
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.01, AS04.02A-R4, AS04.08
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully executed the `pivSelectCardApplication` command.
Test Steps	1. Set cardHandle := <<valid cardHandle>> 2. Set keyReference := <<'9B'>> 3. Set algorithmIdentifier := <<'00', '03', '08', '0A', or '0C'>> 4. Set algorithmInput := <<Use the Dynamic Authentication Template format (Table 7 of SP 800-73-4 Part 2) to request a witness from the card, then issue a second call that contains the decryption of the encrypted challenge from the card appended with the client's application-generated challenge.>> 5. Create algorithmOutput reference 6. Call pivCrypt with • *(IN)* cardHandle • *(IN)* algorithmIdentifier • *(IN)* keyReference • *(IN)* algorithmInput • *(IN) inputLength* • *(OUT)* algorithmOutput • *(INOUT) outputLength*
Expected Result(s)	1. The first call returns with status_word of `PIV_OK` with the `algorithmOutput` carrying the encrypted challenge from the card. 2. The second call returns with status_word of `PIV_OK` with `algorithmOutput` carrying the encrypted data of the client's application-generated challenge.
Postcondition(s)	The client application and the PIV Card Application are mutually authenticated and set security state accordingly.

B.9.1.3 Authenticate the Client Application to Card Application

Purpose	Exercise the PIV Card Application to perform External Authenticate.

Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.01, AS04.02A-R4, AS04.08
Precondition(s)	1. The client application owns a connection to the Card Application accessible through `cardHandle`. 2. The client application has successfully executed the `pivSelectCardApplication` command.
Test Steps	1. Set cardHandle := <<valid cardHandle>> 2. Set keyReference := <<'9B'>> 3. Set algorithmIdentifier := <<'00', '03', '08', '0A', or '0C'>> 4. Set algorithmInput := <<Use the Dynamic Authentication Template format (Table 7 of SP 800-73-4 Part 2) to request a challenge and then to encode an encrypted response in the next call>> 5. Create algorithmOutput reference 6. Call pivCrypt with • *(IN)* cardHandle • *(IN)* algorithmIdentifier • *(IN)* keyReference • *(IN)* algorithmInput • *(IN) inputLength* • *(OUT)* algorithmOutput • *(INOUT) outputLength*
Expected Result(s)	1. The first call returns with status_word of `PIV_OK` with the `algorithmOutput` carrying the challenge from the card. 2. The second call returns the status_word of `PIV_OK`.
Postcondition(s)	The client application is authenticated to the PIV Card Application. The PIV Card Application updated its application security status.

B.9.2 Test Assertions for Error Conditions

B.9.2.1 Identify and Handle an Invalid cardHandle

Purpose	Ensure the PIV Middleware can detect invalid card handles.
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the PIV Card Application.
Test Steps	1. Set cardHandle := <<an invalid cardHandle>> 2. Set keyReference := <<a recognized key reference>> 3. Set algorithmIdentifier := <<a recognized Algorithm Identifier>> 4. Set algorithmInput := <<byte sequence compatible with the chosen algorithm identifier AND keyReference>> 5. Create algorithmOutput reference

	6. Call pivCrypt with • *(IN)* cardHandle • *(IN)* algorithmIdentifier • *(IN)* keyReference • *(IN)* algorithmInput • *(IN) inputLength* • *(OUT)* algorithmOutput • *(INOUT) outputLength*
Expected Result(s)	Call returns with status_word of PIV_INVALID_CARD_HANDLE.
Postcondition(s)	The precondition states are unaffected.

B.9.2.2 Identify and Handle an Invalid keyReference or Algorithm

Purpose	Ensure the PIV Middleware detects invalid key references or algorithms.
Target	pivCrypt
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08
Precondition(s)	1. The client application owns a connection to the Card Application accessible through cardHandle. 2. The client is logged into the PIV Card Application.
Test Steps	1. Set cardHandle := <<a valid cardHandle>> 2. Either the keyReference or algorithmIdentifier, or both, set to an invalid value. 3. Set algorithmInput := <<byte sequence compatible with a type of authentication encoded according to the format in the Dynamic Authentication Template – Table 7 of SP 800-73-4 Part 2>> 4. Create algorithmOutput reference 5. Call pivCrypt with • *(IN)* cardHandle • *(IN)* algorithmIdentifier • *(IN)* keyReference • *(IN)* algorithmInput • *(IN) inputLength* • *(OUT)* algorithmOutput • *(INOUT) outputLength*
Expected Result(s)	Call returns with status_word of PIV_INVALID_KEYREF_OR_ALGORITHM.
Postcondition(s)	The precondition states are unaffected.

B.9.2.3 Identify and Handle the keyReference Set to the PIV Secure Messaging Key

Purpose	Ensure the PIV Middleware detects and handles a reference to the PIV Secure Messaging key.
Target	pivCrypt
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08, AS04.08A-R4

Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Set keyReference := <<'04'>>`
	`3. Set algorithmIdentifier := <<'27' or '2E'>>`
	`4. Set algorithmInput := <<byte sequence compatible with the type of authentication (see Section 4.1.8 of SP 800-73-4, Part 2) encoded according to the format in the Dynamic Authentication Template – Table 7 of SP 800-73-4 Part 2>>`
	`5. Create algorithmOutput reference`
	`6. Call pivCrypt with`
	• *(IN)* `cardHandle`
	• *(IN)* `algorithmIdentifier`
	• *(IN)* `keyReference`
	• *(IN)* `algorithmInput`
	• *(IN) inputLength*
	• *(OUT)* `algorithmOutput`
	• *(INOUT) outputLength*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_KEYREF_OR_ALGORITHM`.
Postcondition(s)	The precondition states are unaffected.

B.9.2.4 Identify and Handle an Invalid Input Data

Purpose	Ensure that the PIV Middleware identifies and handles input data (`algorithmInput`) that is not compatible with the requested algorithm/key combination.
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client is logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<a valid cardHandle>>`
	`2. Set keyReference := <<a key reference compatible with the algorithmIdentifier input value>>`
	`3. Set algorithmIdentifier := <<a recognized Algorithm Identifier>>`
	`4. Set algorithmInput := <<byte sequence not compatible with the type of authentication and not encoded according to the format in the Dynamic Authentication Template – Table 7 of SP 800-73-4 Part 2>>`
	`5. Create algorithmOutput reference`
	`6. Call pivCrypt with`
	• *(IN)* `cardHandle`
	• *(IN)* `algorithmIdentifier`
	• *(IN)* `keyReference`
	• *(IN)* `algorithmInput`
	• *((IN) inputLength*

	• *(OUT)* `algorithmOutput` • *(INOUT) `outputLength`*
Expected Result(s)	Call returns with status_word of `PIV_INPUT_BYTES_MALFORMED`.
Postcondition(s)	1. The PIV Card Application returns to the state it had prior to the `pivCrypt` function call. 2. The precondition states are unaffected.

B.9.2.5 Security Conditions are Enforced

Purpose	Verify that Internal Authenticate is performed with enforced security conditions with/without logging into (PIN VERIFY) the PIV Card Application (see Table 4, Part 1 and Table 2, Part 2 (GENERAL AUTHENTICATE) security condition requirements).
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully executed the `pivSelectCardApplication` command. 3. The client is not logged into the PIV Card Application.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<'9A'>>`
	`3. Set algorithmIdentifier := <<'07' or '11'>>`
	`4. Set algorithmInput : = <<Use the Dynamic Authentication Template format (Table 7 of SP 800-73-4 Part 2) to encode a challenge to be sent to the card>>`
	`5. Create algorithmOutput reference`
	`6. Call pivCrypt with`
	• *(IN)* `cardHandle`
	• *(IN)* `algorithmIdentifier`
	• *(IN)* `keyReference`
	• *(IN)* `algorithmInput`
	• *(IN) `inputLength`*
	• *(OUT)* `algorithmOutput`
	• *(INOUT) `outputLength`*
	`7. Repeat steps 1 – 6 but with the '9E' key reference (Card Authentication key) and algorithmIdentifier := <<'00', '03', '07', '08', '0A', '0C' or '11'>>`
	`8. Repeat steps 1 – 6, but with the '9C' key reference (digital signature key) and algorithmIdentifier <<'07', '11' or '14'>>`
	`9. Repeat steps 1 – 6, but with the '9D' key reference (key management key) and algorithmIdentifier << '07', '11' or '14'>>`

Expected Result(s)	1. Step 7 call returns with status_word of `PIV_OK` with the `algorithmOutput` carrying the encrypted challenge from the card.

	2. All other calls return with status_word of `PIV_SECURITY_CONDITIONS_NOT_SATISFIED`.
Postcondition(s)	1. The PIV Card Application returns to the state it had prior to the `pivCrypt` function call. 2. The precondition states are unaffected.

B.9.2.6 Identify and Handle an Insufficient Buffer

Purpose	Ensure that the PIV Middleware can identify and handle when it has been provided an insufficient length for the algorithm output for the `pivCrypt`.
Target	`pivCrypt`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.3.1 2. AS04.08B-R4
Precondition(s)	1. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 2. The client application has successfully selected the PIV Card Application. 3. Length of the buffer allocated for data by the client application is only 1 byte.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Set keyReference := <<'9E'>>`
	`3. Set algorithmIdentifier := <<'07' or '11'>>`
	`4. Set algorithmInput := <<byte sequence compatible with the chosen algorithm identifier and keyReference>>`
	`5. Create algorithmOutput reference set to 1`
	`6. Call pivCrypt with`
	• *(IN)* `cardHandle`
	• *(IN)* `algorithmIdentifier`
	• *(IN)* `keyReference`
	• *(IN)* `algorithmInput`
	• *(IN) inputLength*
	• *(OUT)* `algorithmOutput`
	• *(INOUT) outputLength*

Expected Result(s)	Call returns with status_word of `PIV_INSUFFICIENT_BUFFER` and sets the `outputLength` to the length of the algorithm output.
Postcondition(s)	The precondition states are unaffected.

B.10 pivMiddlewareVersion

B.10.1 Valid Test Assertions

B.10.1.1 Retrieve the Supported PIV MiddlwareVersion

Purpose	Ensure the PIV Middleware provides its version number to the client application.

Target	`pivMiddlewareVersion`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.1.1 2. AS04.01, AS04.02A-R4, AS04.03B-R4
Precondition(s)	N/A
Test Steps	`1. Call pivMiddlewareVersion with` • *`(OUT) version`*
Expected Result(s)	Function call returns with the version string "800-73-4 Client API" or "800-73-4 Client API with SM" if secure messaging is supported.
Postcondition(s)	N/A

B.11 pivEstablishSecureMessaging

The following tests shall be performed using a PIV Card that has implemented support for the virtual contact interface (VCI) and that has been configured to require submission of a pairing code in order to establish the VCI. These tests need to be performed using a card that has been configured to require submission of a pairing code in order to test the ability of the PIV Middleware to submit the pairing code over secure messaging when the client application calls pivLogIntoCardApplication with a pairing code. It is not necessary to also test the ability of the PIV Middleware to work with a card that is configured to not require the submission of a pairing code in order to establish the VCI since the middleware performs the same steps for each function regardless of whether the PIV Card is configured to require the pairing code.

B.11.1 Valid Test Assertions

B.11.1.1 Establish Secure Messaging

Purpose	Ensure the PIV Middleware can establish secure messaging session with the PIV Card Application.
Target	`pivEstablishSecureMessaging`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.2 2. AS04.07A-R4
Precondition(s)	1. A valid PIV Card is placed within the reading range of the contactless reader. 2. There exists a valid connection between the test system and an instance of the contactless reader. 3. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 4. The client application has successfully selected the PIV Card Application. 5. No other contactless card is within the proximity of the reader.

Test Steps	`1. Set cardHandle := <<valid cardHandle>>`
	`2. Call pivEstablishSecureMessaging with`
	• *`(IN) cardHandle`*

Expected Result(s)	Call returns with status_word of `PIV_OK`.
Postcondition(s)	Secure messaging session is established.

B.11.1.2 Command Execution with Established Secure Messaging

Purpose	Ensure the PIV Middleware implements ACRs across SM with VCI and pairing code.
Target	`pivEstablishSecureMessaging`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.2 2. AS04.07A-R4, AS04.07B-R4
Precondition(s)	1. A valid PIV Card is placed within the reading range of the contactless reader. 2. There exists a valid connection between the test system and an instance of the contactless reader. 3. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 4. The client application has successfully selected the PIV Card Application. 5. No other contactless card is within the proximity of the reader. 6. Secure messaging has been established by calling the `pivEstablishSecureMessaging` function.

Test Steps	`1. Call pivLogIntoCardApplication with '98'`
	• `(IN) cardHandle`
	• `(IN) authenticators`
	• `(IN) AuthLength`
	`2. Call pivLogIntoCardApplication with ('00' or '80')`
	• `(IN) cardHandle`
	• `(IN) authenticators`
	• `(IN) AuthLength`
	`3. Call pivGetData with (Cardholder Fingerprints, Facial Image, Iris Images, Printed Information, and the Pairing Code Reference Data Container)`
	• `(IN) cardHandle`
	• `(IN) OID`
	• `(IN) oidLength`
	• `(OUT) data`
	• `(INOUT) DataLength`
	`4. Call pivCrypt with ('9A', '9D', and all retired key management keys ('82'-'95') located on the card)`
	• `(IN) cardHandle`
	• `(IN) algorithmIdentifier`
	• `(IN) keyReference`
	• `(IN) algorithmInput`
	• `(IN) inputLength`
	• `(OUT) algorithmOutput`
	• `(INOUT) outputLength`

	`5. Perform pivLogIntoCardApplication from step 2 and repeat step 4 using the '9C' key`
Expected Result(s)	Each function call returns the status_word `PIV_OK`.
Postcondition(s)	The security status of the pairing code and PIN used in steps 1 and 2 are set to TRUE.

B.11.2 Test Assertions for Error Conditions

B.11.2.1 Identify and Handle an Invalid cardHandle

Purpose	Ensure the PIV Middleware detects invalid card handles.
Target	`pivEstablishSecureMessaging`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.2 2. AS04.07A-R4
Precondition(s)	1. A valid PIV Card is placed within the reading range of the contactless reader. 2. There exists a valid connection between the test system and an instance of the contactless reader. 3. The client application owns a connection to the PIV Card Application accessible through `cardHandle`. 4. The client application has successfully selected the PIV Card Application. 5. No other contactless card is within the proximity of the reader. 6. Secure messaging has not been established.

Test Steps	`1. Set cardHandle := <<an invalid cardHandle>>`
	`2. Call pivEstablishSecureMessaging with`
	• *`(IN) cardHandle`*

Expected Result(s)	Call returns with status_word of `PIV_INVALID_CARD_HANDLE`.
Postcondition(s)	1. The PIV Middleware remains in the state it had prior to the `pivEstablishSecureMessaging` function call. 2. The precondition states are unaffected.

B.11.2.2 Secure Messaging Failure

Purpose	Ensure the PIV Middleware correctly handles a secure messaging failure.
Target	`pivEstablishSecureMessaging`
Reference(s)	1. SP 800-73-4 Part 3, Section 3.2.2 2. AS04.07A-R4
Precondition(s)	1. A valid PIV Card is placed within the reading range of the contactless reader. 2. There exists a valid connection between the test system and an instance of the contactless reader. 3. The client application owns a shared connection to the PIV Card Application (`sharedConnection := true`).

	4. No other contactless card is within the proximity of the reader. 5. Secure messaging has been successfully established by calling the `pivEstablishSecureMessaging` function.
Test Steps	1. Tester removes the PIV Card from the reading range of the contactless reader so that the card loses power. 2. Tester brings the PIV Card back into range of the contactless reader. 3. Set sharedConnection := true 4. Set connectionDescription := <<valid connection>> 5. Create cardHandle reference 6. Call pivConnect with • *(IN)* sharedConnection • *(INOUT)* connectionDescription • *(INOUT) CDLength* • *(OUT) cardHandle* 7. Call pivSelect with (AID of the PIV Card) • (IN) cardHandle • (IN) applicationAID • (IN) AIDLength • (OUT) applicationProperties • (INOUT) APLength 8. Call pivGetData with (OID of the X.509 Certificate for Card Authentication) • (IN) cardHandle • (IN) OID • (IN) oidLength • (OUT) data • (INOUT) DataLength
Expected Result(s)	1. Step 6 returns with status_word of `PIV_OK` and initialized `cardHandle`. 2. Step 7 returns `PIV_OK` and the initialized application properties reference. 3. Step 8 returns `PIV_SM_FAILED`.
Postcondition(s)	Secure Messaging is not established

Appendix C—Card Command Interface Test Assertions

This appendix specifies the tests that shall be performed on PIV Card Applications. Unless otherwise specified:

- Tests within a subsection titled "Contact Interface" shall be performed over the contact interface of the PIV Card without the use of secure messaging. These tests shall be performed for all PIV Card Applications being tested.
- Tests within a subsection titled "Contactless Interface" shall be performed over the contactless interface of the PIV Card without the use of secure messaging. These tests shall be performed for all PIV Card Applications being tested.
- Tests within a subsection titled "Secure Messaging Interface" shall be performed over the contactless interface of the PIV Card with secure messaging. These tests shall be performed for all PIV Card Applications that support secure messaging. If the Discovery Object is present and Bit 4 of the first byte of the PIN Usage Policy is set to one (indicating that the PIV Card Application has implemented the optional VCI), then Bit 3 of the first byte of the PIN Usage Policy shall be set to zero for these tests (to indicate that the pairing code is required to establish the VCI).
- Tests within a subsection titled "Virtual Contact Interface" shall be performed over the contactless interface of the PIV Card with secure messaging. These tests shall be performed for all PIV Card Applications that support the VCI. For these tests, the Discovery Object shall be present and Bit 4 of the first byte of the PIN Usage Policy shall be set to one. The tests shall be run with Bit 3 of the first byte of the PIN Usage Policy set to zero and with the pairing code having been submitted to the PIV Card Application. If the PIV Card Application also supports setting Bit 3 of the first byte of the PIN Usage Policy to one, then these tests shall additionally be performed in that configuration with the security status indicator of the pairing code set to FALSE.

Test Assertion Template

Purpose	A quick description of the test and why it is being run.
Reference(s)	1. References to SP 800-73-4 or other relevant publications. 2. References to DTRs.
Precondition(s)	Anything that must be done or known prior to executing the scenario.
Test Scenario	`Sequence of APDU calls.`
Expected Result(s)	What the expected execution path yields in terms of progress and values.
Postcondition(s)	A description of the card application state once the test scenario completes.

Note: The status words returned in all SM and VCI interface tests refer to the BER-TLV status words associated with the card commands, not the secure messaging SW protocol status words.

C.1 Card Commands for Data Access

C.1.1 SELECT Card Command

C.1.1.1 Contact Interface

Purpose	Validates that the PIV Card executes the SELECT card command through the contact interface for the following conditions: 1. Long AID. 2. Right-truncated short AID.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.1.1 2. AS01.04, AS01.05, AS01.06, AS01.07, AS01.08, AS03.02, AS05.01, AS05.05, AS05.06, AS05.07, AS05.08, AS05.09, AS05.10, AS05.11
Precondition(s)	1. The IUT (i.e., the PIV Card that is the subject of the test) is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. No application is currently connected to the PIV Card Application.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Send SELECT card command without the version number`
	`• AID == 'A0 00 00 03 08 00 00 10 00'`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. Check that the application property template conforms to Table 3 of SP 800-73-4 Part 2. 2. The command returns the application property template with the status word '90 00' at the end. Check that the application property template conforms to Table 3 of SP 800-73-4 Part 2.
Postcondition(s)	PIV Card Application is now the currently selected application. The application security status of the PIV Card Application is established.

C.1.1.2 Error Condition

Purpose	Validates that the PIV Card Application is not deselected while the currently selected application is the PIV Card Application and the SELECT command is sent with an AID that is not supported by the card.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.1.1 2. AS05.10
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. No application is currently connected to the PIV Card Application.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Send VERFY card command with`
	`• P2, key reference value is set to '80'`
	`• Data file of the command will contain the PIN value obtained from the vendor, padded with 'FF' to complete to total length of the value to 8 bytes.`
	`3. Repeat step 1 with`
	`• AID == 'A0 00 00 03 08 00 00 00 00' (invalid AID)`
	`4. Send GET DATA card command with`
	`• Data field of the command containing the tag of the Cardholder Fingerprints data object`
	`5. Repeat step 1 with`
	`• AID == 'A0 00 00 03 08 00 00 10 00'`
	`6. Repeat steps 3 and 4`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The command returns the status word '90 00'. 3. The command returns '6A 82' (application not found) 4. The command returns the Cardholder Fingerprints object with the status word '90 00' at the end. 5. The command returns the application property template with the status word '90 00' at the end. 6. The commands return the same results as in steps 3 and 4.
Postcondition(s)	The PIV Card Application continues to be the currently selected application and the application security status of the PIV Card Application remains unchanged.

C.1.1.3 Contactless Interface

Purpose	Validates conformance of the SELECT card command through the contactless interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.01, AS05.05, AS05.07, AS05.08, AS05.10
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Send SELECT card command without the version number`
	`• AID == 'A0 00 00 03 08 00 00 10 00'`
	`3. Repeat step 1 with`
	`• AID == 'A0 00 00 03 08 00 00 00 00' (invalid AID)`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. The application property template conforms to Table 3 of SP 800-73-4 Part 2. 2. The command returns the application property template with the status word '90 00' at the end. The application property template conforms to Table 3 of SP 800-73-4 Part 2. 3. The command returns '6A 82' (application not found).
Postcondition(s)	PIV Card Application is the currently selected application.

C.1.2 GET DATA card command

C.1.2.1 Contact Interface

Purpose	Validates that the PIV Card accepts the GET DATA command through the contact interface and with the access rule of each container as specified in Table 2 of SP 800-73-4 Part 1. This test is applicable to the mandatory data objects required by SP 800-73-4, and the optional data objects, when supported by the card.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.1.2 2. AS02.03, AS05.01, AS05.12, AS05.12A, AS05.02
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. No application is currently connected to the PIV Card Application. 4. The optional containers supported by the card are recorded.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Send GET DATA command with`
	`• Data field of the command containing the tag of the Card Capability Container data object`
	`3. Send GET DATA command with`
	`• Data field of the command containing the tag of the CHUID data object`
	`4. Send GET DATA command with`
	`• Data field of the command containing the tag of the X.509 Certificate for PIV Authentication data object`
	`5. Send GET DATA command with`
	`• Data field of the command containing the tag of the Cardholder Fingerprints data object`
	`6. Send GET DATA command with`
	`• Data field of the command containing the tag of the Security Object`
	`7. Send GET DATA command with`
	`• Data field of the command containing the tag of the Cardholder Facial Image data object`

	8. Send GET DATA command with • Data field of the command containing the tag of the X.509 Certificate for Card Authentication data object 9. Send GET DATA command with • Data field of the command containing the tag of the X.509 Certificate for Digital Signature data object 10. Send GET DATA command with • Data field of the command containing the tag of the X.509 Certificate for Key Management data object 11. If Printed Information data object is supported, send GET DATA command with • Data field of the command containing the tag of the Printed Information data object 12. If Cardholder Iris images data object is supported, send GET DATA command with • Data field of the command containing the tag of the Cardholder Iris images data object 13. If Discovery Object data object is supported, send GET DATA command with • Data field of the command containing the tag of the Discovery Object data object 14. If retired keys are supported on card: A) Send GET DATA command with • Data field of the command containing the tag of the Key History Object data object B) For each implemented (up to twenty) Retired X.509 Certificate for Key Management • Send GET DATA command with the data field of the command containing the tag of a Retired X.509 Certificate for Key Management data object 15. If OCC is supported, send GET DATA command with • Data field of the command containing the tag of the Biometric Information Templates Group Template data object 16. If secure messaging for non-card-management opertions is supported, send GET DATA command with • Data field of the command containing the tag of the Secure Messaging Certificate Signer data object 17. If the virtual contact interface is supported, send GET DATA command with • Data field of the command containing the tag of the Pairing Code Reference Data Container data object 18. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain the PIN value obtained from the vendor, padded with 'FF' to complete the total length of the value to 8 bytes • This command (and steps 19 – 23) shall additionally be executed with: 1) P2 = '00' if the card supports the Global PIN (as indicated by the PIN Usage Policy within the Discovery Object) and 2) P2='96' and '97' if the card supports OCC and it satisfies the PIV ACRs. For key reference '00' the subsequent steps have the same expected result as when the key reference '80' is used.

	`NOTE: If multiple key references are being tested the security status of the card will need to be reset before a subsequent key reference can be tested`
	`19. Send GET DATA command with`
	• `Data field of the command containing the tag of the Cardholder Fingerprints data object`
	`20. If Printed Information data object is supported, send GET DATA command with`
	• `Data field of the command containing the tag of the Printed Information data object`
	`21. Send GET DATA command with`
	• `Data field of the command containing the tag of the Cardholder Facial Image data object`
	`22. If Cardholder Iris Images data object is supported, send GET DATA command with`
	• `Data field of the command containing the tag of the Cardholder Iris Images data object`
	`23. If the virtual contact interface is supported, send GET DATA command with`
	• `Data field of the command containing the tag of the Pairing Code Reference Data Container data object`
	`24. Send GET DATA command with`
	• `Data field of the command containing a tag that does not identify any of the data objects resident on the card`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. For steps 2, 3, 4, 6, 8, 9, 10, 13, 14A, 14B, 15, and 16 the command returns the requested data object along with the status word '90 00' at the end. 3. For steps 5, 7, and 12 the command returns '69 82' (security status not satisfied), due to lack of PIN entry. 4. For steps 11 and 17 the command returns '69 82' (security status not satisfied), due to lack of PIN entry or successful OCC. 5. In step 18, the command returns the status word '90 00' 6. For steps 19, 21, and 22 the command returns: • The requested data object along with the status word '90 00' at the end, if step 18 was performed using key reference '00' or '80' • Status word '69 82' (security status not satisfied), if step 18 was performed using key reference '96' or '97'. 7. For steps 20 and 23 the command returns the requested data object along with the status word '90 00' at the end. 8. In step 24, the command returns '6A 82' (data object not found).
Postcondition(s)	N/A

C.1.2.2 Contactless Interface

Purpose	Validates the conformance of the GET DATA command through the contactless interface. This test is applicable to the mandatory data objects required by Table 2 of SP 800-73-4 Part 1, and the optional data objects when supported by the card.
Reference(s)	1. Table 2 of SP 800-73-4 Part 2 2. AS05.01, AS05.02, AS05.12, AS05.12A
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader.
Test Scenario	`1. Repeat the steps 1-17 and 24 of test C.1.2.1`
Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. In steps 2, 4, 5, 6, 7, 9, 10, 11, 12, 14A, 14B, and 17 the command returns the status word '69 82' (security status is not satisfied), due to the contactless interface. 3. In steps 3, 8, 13, 15, and 16 the command returns the requested data object along with the status word '90 00' at the end. 4. In step 24, the command returns '6A 82' (data object not found).
Postcondition(s)	N/A

C.1.2.3 Secure Messaging Interface

Purpose	Validates the conformance of the GET DATA command using secure messaging. This test is applicable to the mandatory data objects required by Table 2 of SP 800-73-4 Part 2, and the optional data objects when supported by the card.
Reference(s)	1. Table 2 of SP 800-73-4 Part 2 2. AS05.01, AS05.02, AS05.12, AS05.12A
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. Secure messaging keys have been established and secure messaging is used in the test scenario.

Test Scenario	`1. Repeat step 1 from Test C.1.2.1 without secure messaging.`
	`2. Repeat the steps 2-17 and 24 from the Test C.1.2.1 using the '0C' CLA byte`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end.

	2. In steps 2, 4, 5, 6, 7, 9, 10, 11, 12, 14A, 14B, and 17 the command returns status word '69 82' (security status is not satisfied). 3. In steps 3, 8, 13, 15, and 16, the command returns the requested data object along with the status word '90 00' at the end. 4. In step 24, the command returns '6A 82' (data object not found).
Postcondition(s)	N/A

C.1.2.4 Virtual Contact Interface

Purpose	Validates the conformance of the GET DATA command through the VCI. This test is applicable to the mandatory data objects required by Table 2 of SP 800-73-4 Part 2, and the optional data objects when supported by the card.
Reference(s)	1. Table 2 of SP 800-73-4 Part 2 2. AS05.01, AS05.02, AS05.12, AS05.12A
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. There exists a valid VCI connection to the card.

Test Scenario	`1. Repeat step 1 from Test C.1.2.1 without secure messaging.`
	`2. Repeat the steps 2-24 from the Test C.1.2.1 using the '0C' CLA byte`

Expected Result(s)	The expected results are the same as those specified in Test C.1.2.1 for the contact interface.
Postcondition(s)	N/A

C.2 Commands for Authentication

C.2.1 VERIFY Card Command

C.2.1.1 Contact Interface

Purpose	Validates the following conditions associated with the VERIFY command: 1. With an invalid key reference. 2. Successful execution of the command (with PIV Card Application PIN and (if supported) Global PIN and OCC). 3. Execution of the command with a PIN not formatted per SP 800-73-4. 4. Multiple execution of the command with an incorrect PIN (formatted correctly) until the retry counter reaches zero. 5. Reset the security status.

Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.1 2. AS01.16, AS05.01, AS05.12 through AS05.22A, AS05.18A-R4, AS05.22C-R4, and AS05.22D-R4.
Precondition(s)	1. The PIV Card Application PIN and Global PIN (if supported) are each 6 bytes in length. 2. PIV Card Application PIN is recorded. 3. Global PIN (if supported) is recorded. 4. Cardholder fingerprint minutia for on-card comparison is recorded (if OCC is implemented). 5. Pairing code (if supported) is recorded. 6. The reset retry counter values of PIV Card Application PIN, Global PIN (if implemented), and OCC (if implemented) are recorded. 7. The card is not blocked and security status is set to FALSE for all authenticators.
Test Scenario	1. Send SELECT card command with • AID == 'A0 00 00 03 08 00 00 10 00 01 00' 2. Send GET DATA command with • Data field of the command containing the tag of the Discovery Object. Parse the PIN Usage Policy to discover the PIN and OCC supported by the card. Perform the test for all cases that match the PIN Usage Policy 2a. Test case for the mandatory PIV Card Application PIN ('80') 1. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 2. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIV Card Application PIN value, padded with 'FF' to complete the total length of the value to 8 bytes 3. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 4. Reset the security status of the '80' key reference by sending the VERIFY command with • P2, key reference value is set to '80' • P1 parameter is 'FF' and both L_c and the data field are absent 5. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 6. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIV Card Application PIN value, NOT padded with 'FF', so that the total length of the value is less than 8 bytes 7. Send VERIFY card command with

	• P2, key reference value is set to '80' • Data field of the command will contain the correct PIV Card Application PIN value, padded with 'FF' to complete the total length of the value to 10 bytes 8. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIV Card Application PIN value, padded to 8 bytes with 'FF' in byte 7 and 'AA' in byte 8 9. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain an arbitrary 6-digit PIV Card Application PIN value where the first byte is 0x5A and all other non-padded bytes contain values limited to 0x30 – 0x39, padded with 'FF' to complete the total length of the value to 8 bytes 9a. Repeat step 9 five times with byte positions 2, 3, 4, 5, and 6 containing the 0x5A byte, respectively. Note: It may be necessary to send the VERIFY command with a correct PIV Card Application PIN in order to prevent the retry counter from decrementing to zero. 10. Send VERIFY card command repeatedly, until after the issuer specified maximum number of PIN tries is exceeded with • P2, key reference value is set to '80' • Data field of the command will contain an arbitrary, but correctly formatted, PIN value other than what is obtained from the vendor, padded with 'FF' to complete the total length of the value to 8 bytes 2b. Test case for implementations that support the Global PIN ('00') for PIV data access and command execution 1. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 2. Send VERIFY card command with • P2, key reference value is set to '00' • Data field of the command will contain the correct Global PIN value, padded with 'FF' to complete the total length of the value to 8 bytes 3. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 4. Reset the security status of the '00' key reference by sending the VERIFY command with • P2, key reference value is set to '00' • P1 parameter is 'FF' and both L_c and the data field are absent

	5. Send GET DATA command with • Data field of the command containing the tag of the Cardholder Fingerprints data object 6. Send VERIFY card command with • P2, key reference value is set to '00' • Data field of the command will contain the correct Global PIN value, NOT padded with 'FF' so that the total length of the field is less than 8 bytes 7. Send VERIFY card command with • P2, key reference value is set to '00' • Data field of the command will contain the correct Global PIN, padded with 'FF' to complete the total length of the value to 10 bytes 8. Send VERIFY card command with • P2, key reference value is set to '00' • Data field of the command will contain the correct Global PIN value, padded to 8 bytes with 'FF' in byte 7 and 'AA' in byte 8 9. Send VERIFY card command with • P2, key reference value is set to '00' • Data field of the command will contain an arbitrary 6-digit Global PIN value where the first byte is 0x5A and all other non-padded bytes contain values limited to 0x30 – 0x39, padded with 'FF' to complete the total length of the value to 8 bytes 9a. Repeat step 9 five times with byte positions 2, 3, 4, 5, and 6 containing the 0x5A byte, respectively. Note: It may be necessary send the VERIFY command with a correct Global PIN in order to prevent the retry counter from decrementing to zero. 10. Send VERIFY card command repeatedly until after the issuer specified maximum number of PIN tries is exceeded with • P2, key reference value is set to '00' • Data field of the command will contain an arbitrary, but correctly formatted, PIN value other than what is obtained from the vendor, padded with 'FF' to complete the total length of the value to 8 bytes 2c. Test case for implementations that support OCC ('96' and '97') for PIV data access and command execution. 1. Send VERIFY card command with • P2, key reference value is set to '96' • Data field of the command will contain a value that matches the Primary Finger OCC value 2. Send GET DATA command with • Data field of the command containing the tag of the Printed Information data object

	3. Reset the security status of the '96' key reference by sending the VERIFY command with • P2, key reference value is set to '96' • P1 parameter is 'FF' and both L_c and the data field are absent 4. Send VERIFY card command with • P2, key reference value is set to '97' • Data field of the command will contain a value that matches the Secondary Finger OCC value 5. Send GET DATA command with • Data field of the command containing the tag of the Printed Information data object 6. Reset the security status of the '97' key reference by sending the VERIFY command with • P2, key reference value is set to '97' • P1 parameter is 'FF' and both L_c and the data field are absent 7. Send GET DATA command with • Data field of the command containing the tag of the Printed Information data object 8. Send VERIFY card command with • P2, key reference value is set to '96' • Data field of the command will contain a random fingerprint value. The fingerprint is truncated so that the total length is less than 3 bytes times the minimum number of minutia specified in the BIT Group Template for key reference '96' 9. Send VERIFY card command with • P2, key reference value is set to '96' • Data field of the command will contain a random fingerprint value. The fingerprint is padded so that the total length is more than 3 bytes times the maximum number of minutia specified in the BIT Group Template for key reference '96' 10. Send VERIFY card command repeatedly until after the issuer specified maximum number of OCC tries is exceeded with • P2, key reference value is set to '96' • Data field of the command will contain an arbitrary, but correctly formatted, value that does not match the Primary Finger OCC value obtained from the vendor 11. Repeat steps 8-10 with key reference '97'.
Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The command returns either 1) '6A 82' (data object not found) or 2) the Discovery Object with the status word '90 00' at the end (verify that the returned PIN Usage Policy matches to what is described in vendor documentation. 2a:

	1. The command returns '69 82' (security status not satisfied). 2. The command returns '90 00' (verify that the retry counter is set to reset retry value). 3. The command returns the Cardholder Fingerprints data object along with the status word '90 00'. 4. The command returns '90 00'. 5. The command returns '69 82' (security status not satisfied). 6. The command returns '6A 80' (incorrect parameter in command data field) or '63 CX' (verification failed, with X indicating the number of further allowed retries) (verify the error code supplied matches what is described in vendor documentation). 7. The command returns '6A 80' (incorrect parameter command data field) or '63 CX' (verification failed, with X indicating the number of further allowed retries (verify the error code supplied matches what is described in vendor documentation). 8. The command returns '6A 80' (incorrect parameter command data field) or '63 CX' (verification failed, with X indicating the number of further allowed retries (verify the error code supplied matches what is described in vendor documentation). 9. The command returns '6A 80' (incorrect parameter command data field) or '63 CX' (verification failed, with X indicating the number of further allowed retries) (verify the error code supplied matches what is described in vendor documentation). 9a. The command returns '6A 80' (incorrect parameter command data field) or '63 CX' (verification failed, with X indicating the numer of further allowed retries (verify the error code supplied matches what is described in vendor documentation). 10. The command returns: • '63 CX' until the maximum number of PIN tries is reached (X indicates the number of further allowed retries). • '69 83' (authentication method blocked) when the maximum number of PIN tries is exceeded. 2b: 1. Steps 1-10 have the same command response as in 2a (1-10). 2c: 1. The command returns '90 00' (verify that the retry counter is set to reset retry value).

	2. The command returns the Printed Information data object along with the status word '90 00'. 3. The command returns the status word '90 00'. 4. The steps 4-6 have the same responses as steps 1-3. 5. In step 7, the command returns '69 82' (security status not satisfied). 5. In steps 8 and 9, the command returns '6A 80' (incorrect parameter command data field). 7. In step 10, the command returns: • '63 CX' until the maximum number of OCC tries is reached (X indicates the number of further allowed retries). • '69 83' (authentication method blocked) when the maximum number of OCC tries is exceeded. 8. In step 11, the repeated steps have the same responses as when performed with key reference '96'.
Postcondition	The card is blocked.

C.2.1.2 Contactless Interface

Purpose	Validates that the PIV Card does not accept the VERIFY command through the contactless interface when secure messaging is not used.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03, AS05.04, AS05.13, AS05.15
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. Cardholder fingerprint minutia is recorded (if OCC is implemented). 5. Pairing code (if supported) is recorded. 6. The reset retry counter values of PIV Card Application PIN, Global PIN (if implemented), OCC (if implemented) are recorded.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Send GET DATA command with`
	`• Data field of the command containing the tag of the Discovery Object. Parse the PIN Usage Policy to discover the PIN, pairing code, and OCC supported by the card. Perform the test for all cases that match the PIN Usage Policy`
	`2a. Test case for the mandatory PIV Card Application PIN ('80')`
	`1. Send VERIFY card command with`

	• `P2, key reference value is set to '80'`
	• `Data field of the command will contain the correct cardholder PIV Card Application PIN value, and padded with 'FF' to complete the total length of the value to 8 bytes`
	`2b. Test case for implementations that support the Global PIN ('00') for PIV data access and command execution as indicated by the Discovery Object's PIN Usage Policy.`
	`1. Send VERIFY card command with`
	• `P2, key reference value is set to '00'`
	• `Data field of the command will contain the correct Global PIN value, and padded with 'FF' to complete the total length of the value to 8 bytes`
	`2c. Test case for implementations that support OCC ('96' and '97') for PIV data access and command execution as indicated by the Discovery Object's PIN Usage Policy`
	`1. Send VERIFY card command with`
	• `P2, key reference value is set to '96'`
	• `Data field of the command will contain a value that matches the Primary Finger OCC value`
	`2. Send VERIFY card command with`
	• `P2, key reference value is set to '97'`
	• `Data field of the command will contain a value that matches the Secondary Finger OCC value`
	`2d. Test case for implementations that support pairing code ('98')`
	`1. Send VERIFY card command with`
	• `P2, key reference value is set to '98'`
	• `Data field of the command will contain the correct pairing code value`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The command returns either 1) the Discovery Object with the status word '90 00' at the end or 2) '6A 82' (data object not found) (verify that the return PIN Usage Policy matches that described in vendor documentation. 2a: 1. The command returns an error status word. 2b: 1. The command returns an error status word. 2c: 1. In the steps 1-2: The command returns an error status word. 2d: 1. The command returns an error status word.
Postcondition(s)	N/A

C.2.1.3 Secure Messaging Interface

Purpose	Validates that only the paring code and OCC succeed in the VERIFY command through the contactless interface using secure messaging when a VCI has not been established.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. SP 800-73-4 Part 1, Table 4 3. AS05.03, AS05.04, AS05.13A-R4, AS05.13B-R4, AS05.15, and AS05.16A-R4
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. The PIV Card Application is the currently selected application. 5. Secure messaging session keys have been established and secure messaging is used in the test scenario.

Test Scenario	`NOTE: set CLA byte to '0C' for all commands to ensure they are sent over secure messaging.`
	`1. Send GET DATA command with`
	  `• Data field of the command containing the tag of the Discovery Object. Parse the PIN Usage Policy to discover the PIN, Pairing Code and OCC supported by the card. Perform the test for all cases that match the PIN Usage Policy`
	`2. Perform the test for all cases that match the PIN Usage Policy`
	  `2a. Test case for the mandatory PIV Card Application PIN ('80')`
	    `1. Perform step 2a in C.2.1.2`
	  `2b. Test case for implementations that support the Global PIN ('00') for PIV data access and command execution as indicated by the Discovery Object’s PIN Usage Policy`
	    `1. Perform step 2b in C.2.1.2`
	  `2c. Test case for implementations that support OCC ('96' and '97') for PIV data access and command execution as indicated by the Discovery Object’s PIN Usage Policy`
	    `1. Send VERIFY card command with`
	      `• P2, key reference value is set to '96'`
	      `• Data field of the command will contain a value that matches the Primary Finger OCC value`
	    `2. Send VERIFY card command with`
	      `• P2, key reference value is set to '97'`
	      `• Data field of the command will contain a value that matches the Secondary Finger OCC value`
	    `3. Send VERIFY card command with`
	      `• P2, key reference value is set to '80'`
	      `• Data field of the command will contain the correct cardholder PIV Card Application PIN`

	value, and padded with 'FF' to complete the total length of the value to 8 bytes 2d. Test case for implementations that support the pairing code ('98') as indicated by the Discovery Object's PIN Usage Policy 1. Send VERIFY card command with • P2, key reference value is set to '98' • Data field of the command will contain an 8 byte random, but correctly formatted, pairing code value 2. Send GET DATA command with • Data field of the command containing the tag of the Card Capability Container 3. Send VERIFY card command with • P2, key reference value is set to '98' • Data field of the command will contain a short 6-byte random pairing code value 4. Send GET DATA command with • Data field of the command containing the tag of the Card Capability Container 5. Send VERIFY command with • P2, key reference value is set to '98' • Data field of the command will contain an arbitrary pairing code with length of 10 bytes 6. Send GET DATA command with • Data field of the command containing the tag of the Card Capability Container 7. Send VERIFY card command with • P2, key reference value is set to '98' • Data field of the command will contain the correct 8 byte pairing code 8. Send GET DATA command with • Data field of the command containing the tag of the Card Capability Container 9. Reset the security status of the '98' key reference by sending the VERIFY command with • P2, key reference value is set to '98' • P1 parameter is 'FF' and both L_c and the data field are absent 10. Send GET DATA command with • Data field of the command containing the tag of the Card Capability Container
Expected Result(s)	1. The command returns either 1) the Discovery Object with the status words '90 00' at the end or 2) '6A 82' (data object not found) (verify that the return PIN Usage Policy matches that described in vendor documentation. 2a: 1. The command returns an error status word. 2b: 1. The command returns an error status word. 2c:

	1. The command returns '90 00' (verify that the retry counter is set to reset retry value). 2. The command returns '90 00' (verify that the retry counter is set to reset retry value). 3. The command returns an error status word. 2d: 1. The command returns '63 00' (verification failed). 2. The command returns '69 82' (security status not satisfied). 3. The command returns either '63 00' (verification failed) or '6A 80' (incorrect parameter in command data field). 4. The command returns '69 82' (security status not satisfied). 5. The command returns either '63 00' (verification failed) or '6A 80' (incorrect parameter in command data field). 6. The command returns '69 82' (security status not satisfied). 7. The command returns '90 00'. 8. The command returns the Card Capability Container object along with the status word '90 00'. 9. The command returns '90 00'. 10. The command returns '69 82' (security status not satisfied).
Postcondition(s)	N/A

C.2.1.4 Virtual Contact Interface

Purpose	Verify the behavior of the VERIFY command under VCI is identical to the contact interface.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.1 2. SP 800-73-4 Part 1, Table 4 3. AS01.17, AS05.01, and AS05.12 through AS05.22A
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. The PIV Card Application is the currently selected application. 5. PIV Card Application PIN is recorded. 6. Global PIN (if supported) is recorded. 7. Cardholder fingerprint minutia for on-card comparison is recorded (if OCC is implemented). 8. The reset retry counter value(s) (maximum number of tries allowed) of the PIV Card Application PIN, Global PIN (if implemented), OCC (if implemented) are recorded. 9. There exists a valid VCI connection to the card.
Test Scenario	`Repeat steps 2, 2a, 2b, 2c from the Test C.2.1.1 using the '0C' CLA byte`

Expected Result(s)	The results from this test have the same command responses as in C.2.1.1 Steps 2, 2a, 2b and 2c, respectively.
Postcondition	The card is blocked.

C.2.2 CHANGE REFERENCE DATA card command

C.2.2.1 Contact Interface

Purpose	Validates that the PIV Card executes the CHANGE REFERENCE DATA command for the following conditions: 1. Without the proper security condition (PIV Card Application PIN and (if supported) Global PIN). 2. After the security condition is satisfied. 3. With an incorrect PIN until the retry counter reaches zero. 4. Verify that the CHANGE REFERENCE DATA command does not change OCC reference data or the pairing code. 5. Ensure that length and format of PIN data is enforced.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.2, 2. AS05.01, AS05.23 through AS05.27
Precondition(s)	1. The reset retry counter values (maximum number of PIN tries allowed) of the PIV Card Application PIN and Global PIN (if supported) are recorded. 2. PIV Card Application PIN is recorded. 3. Pairing code (if supported) is recorded. 4. OCC (if supported) is recorded. 5. Global PIN (if supported) is recorded. 6. The IUT is inserted into the contact reader. 7. There exists a valid PC/SC connection between the test system and the contact reader. 8. No application is currently connected to the PIV Card Application.

Test Scenario	`1. Send SELECT card command with`
	• `AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Perform step 2 in C.2.1.1 (This step reads the Discovery Object from the card and parses the PIN usage Policy sub-element). Perform the test for all test cases that match the PIN Usage Policy`
	`2a. Test case for the mandatory PIV Card Application PIN ('80')`
	`1. Send CHANGE REFERENCE DATA card command with`
	• `P2, key reference value is set to '80'`
	• `Data field of the command will contain the correct PIN value (PIN 1) obtained from the vendor, concatenated without delimitation with an arbitrary new valid PIN value (PIN 2). Both PINs should be padded with 'FF' to complete the total length of each value to 8 bytes`

	2. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command will contain the new PIN value (PIN 2 from previous step), padded with 'FF' to complete the total length of the value to 8 bytes 3. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIN value (PIN 2) concatenated without delimitation with an arbitrary new PIN value (PIN 3) that is padded to less than 8 bytes 4. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIN value (PIN 2) concatenated without delimitation with an arbitrary new PIN value (PIN 4) that is less than 6 bytes but padded to 8 bytes with 'FF' 5. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIN value (PIN 2) concatenated without delimitation with an arbitrary new PIN value (PIN 5) that is 6 bytes but padded to 8 bytes with 'FF' in byte 7 and 'AA' in byte 8 6. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIN value (PIN 2), concatenated without delimitation with an arbitrary new PIN value that contains 0x5A in the first byte position, all other non-padded bytes contain values limited to 0x30 – 0x39 (PIN 6). Both PINs should be padded with 'FF' to complete the total length of each value to 8 bytes (repeat test five times with byte positions 2, 3, 4, 5, and 6 containing the 0x5A byte, respectively) 7. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain an arbitrary PIN value that contains 0x5A in the first byte position, all other non-padded bytes contain values limited to 0x30 – 0x39 (PIN 7), concatenated without delimitation with a properly formatted new PIN value where all non-padded bytes contain values limited to 0x30 – 0x39 (PIN 8). Both PINs should be padded with 'FF' to complete the total length of each value to 8 bytes. (repeat test five times with byte positions 2, 3, 4, 5, and 6 containing the 0x5A byte, respectively)

	8. Send CHANGE REFERENCE DATA card command repeatedly until after the issuer specified maximum number of PIN tries is exceeded with • P2, key reference value is set to '80' • Data field of the command will contain an incorrect PIN value (anything other than PIN 2), concatenated without delimitation with an arbitrary new PIN value (PIN 9). Both PINs should be padded with 'FF' to complete the total length of each value to 8 bytes 2b. Test case for implementations that support the Global PIN ('00') for PIV data access and command execution and the CHANGE REFERENCE DATA command with the Global PIN is implemented with the PIV Card Application. 1. Perform steps 1-8 of 2a using key reference '00' in place of key reference '80' 3. Test case for implementations for which the CHANGE REFERENCE DATA command with the PUK is implemented with the PIV Card Application 1. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '81' • Data field of the command will contain the correct PUK value (PUK 1) concatenated without delimitation with an arbitrary new 8-byte PUK value (PUK 2) 2. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '81' • Data field of the command will contain the correct PUK value (PUK 2) concatenated without delimitation with an arbitrary new 8-byte PUK value (PUK 3) 3. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '81' • Data field of the command will contain the correct PUK value (PUK 3) concatenated without delimitation with an arbitrary new PUK value (PUK 4) that is less than 8 bytes 4. Send CHANGE REFERENCE DATA card command repeatedly until after the issuer specified maximum number of PIN tries is exceeded with • P2, key reference value is set to '81' • Data field of the command will contain an incorrect PUK value (anything other than PUK 3), concatenated without delimitation with an arbitrary new PUK value (PUK 5) 4. Test case for implementations that support OCC 1. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '96' • Data field of the command will contain an arbitrary value of 16 bytes • (repeat test with key reference '97') 5. Test case for implementations that support the pairing code 1. Send CHANGE REFERENCE DATA card command with

	• `P2, key reference value is set to '98'` • `Data field of the command will contain the correct pairing code value (pairing code 1) concatenated without delimitation with an arbitrary new pairing code value (pairing code 2)` `6. Send CHANGE REFERENCE DATA card command with` • `P2, key reference value, is set to a value other than what is supported by the card` • `Data field of the command will contain the correct PIV Card Application PIN value (PIN 2) concatenated without delimitation with an arbitrary new PIN value (PIN 8). Both PINs are truncated or padded with 'FF' to complete the total length of each value to 8 bytes`
Expected Result(s)	1. Command returns the application property template with the status word '90 00' at the end 2. Command returns the same result as in C.2.1.1 2a: 1. The command returns '90 00' (also verify that the retry counter is set to reset retry value). 2. The command returns '90 00'. 3. The command returns '6A 80' (incorrect parameter in command data field) and the retry counter remains unchanged. 4. The command returns '6A 80' (incorrect parameter in command data field) and the retry counter remains unchanged. 5. The command returns '6A 80' (incorrect parameter in command data field) and the retry counter remains unchanged. 6. Each time, the command returns '6A 80' (incorrect parameter in command data field) and the retry counter remains unchanged. 7. Each time, either 1) the command returns '6A 80' and the retry counter remains unchanged or 2) the command returns '63 CX' (X indicates the number of further allowed retries) and the retry counter is decremented. 8. The command returns: • '63 CX' until the maximum number of tries are reached. (X indicates the number of further allowed retries). • '69 83' (reference data change operation blocked) when the maximum number of tries is exceeded. 2b: 1. The expected results in 2b are the same as in 2a. 3.

	1. The command returns '90 00'. 2. The command returns '90 00'. 3. The command returns '6A 80' (incorrect parameter in command data field). 4. The command returns: • '63 CX' until the maximum number of tries are reached. (X indicates the number of further allowed retries) (Verify that first response the value of 'X' is one less than the reset retry value.) • '69 83' (Reference data change operation blocked) when the maximum number of tries is exceeded. 4. The command returns an error status word. 5. The command returns an error status word. 6. The command returns an error status word.
Postcondition(s)	The card is blocked.

C.2.2.2 Contactless Interface

Purpose	Validates that the PIV Card does not accept the CHANGE REFERENCE DATA command through the contactless interface when a VCI has not been established.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03, AS05.24A-R4
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. PIV Card Application PIN is recorded. 5. Global PIN (if supported) is recorded. 6. The reset retry counter value(s) of the PIV Card Application PIN, Global PIN (if implemented) are recorded. 7. The PIN Unblocking Key value is recorded.
Test Scenario	1. Send SELECT card command with • AID == 'A0 00 00 03 08 00 00 10 00 01 00' 2. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '80' • Data field of the command will contain the correct PIN value (PIN 1) obtained from the vendor, concatenated without delimitation with an arbitrary new PIN value (PIN 2). Both PINs should be padded with 'FF' to complete the total length of each value to 8 bytes 3. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '00' • Data field of the command will contain the correct PIN value (PIN 1—if Global PIN is unsupported use an

	arbitrary PIN value) concatenated without delimitation with an arbitrary new PIN value (PIN 2). Both PINs are padded with 'FF' to complete the total length of each value to 8 bytes 4. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '81' • Data field of the command will contain the correct PUK value (PUK 1) concatenated without delimitation with an arbitrary new PUK value (PUK 2). Each PUK is 8 bytes in length 5. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '98' • Data field of the command will contain the correct pairing code value (if pairing code is unsupported use an arbitrary 8-byte pairing code value) obtained from the vendor, concatenated without delimitation with an arbitrary new 8-byte pairing code value. 6. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '96' • Data field of the command will contain an arbitrary value of 16 bytes 7. Send CHANGE REFERENCE DATA card command with • P2, key reference value is set to '97' • Data field of the command will contain an arbitrary value of 16 bytes
Expected Result(s)	1. Command returns the application property template with the status word '90 00' at the end. 2. The command returns an error status word. 3. The command returns an error status word. 4. The command returns an error status word. 5. The command returns an error status word. 6. The command returns an error status word. 7. The command returns an error status word.
Postcondition(s)	PIN remains unchanged.

C.2.2.3 Secure Messaging Interface

Purpose	Validates that the PIV Card does not accept the CHANGE REFERENCE DATA command through the secure messaging interface when a VCI has not been estalished.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03, AS05.24A-R4
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. PIV Card Application PIN is recorded.

	5. Global PIN (if supported) is recorded. 6. The PIV Card Application is the currently selected application on the card. 7. Secure messaging session keys have been established and secure messaging is used in the test scenario.
Test Scenario	`Repeat steps 2-7 of test C.2.2.2 using the '0C' CLA byte`
Expected Result(s)	Commands return the same results as in C.2.2.2.
Postcondition(s)	PIN remains unchanged.

C.2.2.4 Virtual Contact Interface

Purpose	Validates that the PIV Card executes the CHANGE REFERENCE DATA command for the following conditions: 1. Without the proper security condition (PIV Card Application PIN and (if supported) Global PIN). 2. After the security condition is satisfied. 3. With an incorrect PIN until the retry counter reaches zero. 4. Ensure that length and format of PIN data is enforced.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.2 2. AS05.01, AS05.23 through AS05.27, and AS05.24A-R4
Precondition(s)	1. PIV Application PIN and Global PIN (if supported) reset retry counter values (maximum number of PIN tries allowed) are recorded. 2. PIV Card Application PIN is recorded. 3. Global PIN (if supported) is recorded. 4. The IUT is placed within the reading range of the contactless reader. 5. There exists a valid PC/SC connection between the test system and the contactless reader. 6. No other contactless card is within the proximity of the reader. 7. The PIV Card Application is the currently selected application on the card. 8. There exists a valid VCI connection to the card. 9. No application is currently connected to the PIV Card Application.

Test Scenario	`1. Repeat test steps from C.2.2.1, with the exception of steps 1 (selecting PIV Card Application) and 3 (PUK tests), using the '0C' CLA byte`
	`2. Repeat test 4 from C.2.2.2 using the '0C' CLA byte`

Expected Result(s)	1. The results for steps are the same as the results in C.2.2.1. 2. The command returns an error status word.
Postcondition(s)	The card is blocked.

C.2.3 RESET RETRY COUNTER command

C.2.3.1 Contact Interface

Purpose	Validates that the PIV Card executes the RESET RETRY COUNTER command for the following conditions: 1. With the security condition unsatisfied. 2. After the security condition (authenticated with the PUK) is satisfied. 3. With a valid new PIN value not formatted per SP 800-73-4. 4. With a valid new PIN value (formatted correctly). 5. With a valid new PIN value causing the PUK retry counter to be optionally reset. 6. With an unsupported key reference. 7. With the security condition unsatisfied (incorrect PUK) until RESET RETRY COUNTER command is blocked.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.3 2. AS05.01, AS03.07, AS05.30 through AS05.33
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. No application is currently connected to the PIV Card Application. 4. PIV Card Application PIN reset retry counter value (maximum number of PIN tries allowed) is recorded. 5. The value of the counter reference data (PUK) is recorded.
Test Scenario	1. Send SELECT card command with • AID == 'A0 00 00 03 08 00 00 10 00 01 00' 2. Send RESET RETRY COUNTER with • P2, key reference value, is set to '80' • Data field of the command contains the PUK value for key reference '80' concatenated with a PIN value that is not padded to complete 8 bytes 3. Send RESET RETRY COUNTER with • P2, key reference value is set to '80' • Data field of the command contains the PUK value for key reference '80' concatenated with a new PIN value that is 6 bytes but padded to 8 bytes with 'FF' in byte 7 and 'AA' in byte 8 4. Send VERIFY card command with • P2, key reference value is set to '80' • Data field of the command contains an arbitrary, but correctly formatted, PIN value other than what is obtained from the vendor, padded with 'FF' to complete the total length of the value to 8 bytes 5. Send RESET RETRY COUNTER with • P2, key reference value, is set to '80'

- Data field of the command contains the PUK value for key reference '80' concatenated without delimitation with a new PIN (PIN 2) padded with 'FF' to complete the total length of the value to 8 bytes

6. Obtain number of remaining retries of the '80' key reference by sending the VERIFY command with
   - P2, key reference value is set to '80'
   - P1 parameter is '00' and both L_c and the data field are absent
7. Send VERIFY card command with
   - P2, key reference value is set to '80'
   - Data field of the command contains the new PIN (PIN 2) value, padded with 'FF' to complete the total length of the value to 8 bytes

Perform steps 8 - 10 only if the reset of the PIN’s retry counter also resets the PUK retry counter

8. Send RESET RETRY COUNTER with
   - P2, key reference value is set to '80'
   - Data field of the command contains an incorrect PUK value for key reference '80' concatenated without delimitation with a new valid PIN value padded with 'FF' to complete the total length of the value to 8 bytes. (Record the number of remaining retries ‘X’ in return code '63 CX')
9. Repeat step 5
10. Send RESET RETRY COUNTER with
    - P2 key reference value is set to '80'
    - Data field of the command contains an incorrect PUK value for key reference '80' concatenated without delimitation with a new valid PIN value padded with 'FF' to complete the total length of the value to 8 bytes. (Record the number of remaining retries ‘X’ in return code '63 CX')
11. Send RESET RETRY COUNTER with
    - P2, key reference value is set to '80'
    - Data field of the command contains the correct PUK value concatenated without delimitation with an arbitrary PIN value that is less than 6 bytes but padded to 8 bytes with 'FF'
12. Send RESET RETRY COUNTER with
    - P2, key reference value, is set to '80'
    - Data field of the command contains the correct PUK value for key reference '80' concatenated without delimitation with an arbitrary new PIN value that contains 0x5A in the first byte position, all other non-padded bytes contain values limited to 0x30 – 0x39. The new PIN should be padded with 'FF' to complete the total length of the value to 8 bytes (repeat test five times with byte positions 2, 3, 4, 5, and 6 of the PIN containing the 0x5A byte, respectively)
13. Send RESET RETRY COUNTER with
    - P2, key reference value is set to '80'
    - Data field of the command containing an incorrect PUK value concatenated without delimitation with a new PIN

	`padded with 'FF' to complete the total length of the value to 8 bytes. This operation is repeated until the number of resets allowed is exceeded`
Expected Result(s)	1. Command returns the application property template with the status word '90 00' at the end. 2. The command returns '6A 80' (incorrect parameter in command data field). 3. The command returns '6A 80' (incorrect parameter in command data field). 4. The command returns '63 CX' (X == number of retries left) and the retry counter will be decremented by 1. 5. The command returns '90 00'. 6. The command returns '63 CX' (X == number of reties left). Verify that X from this step is > X from step 4. 7. The command returns '90 00'. 8. The command returns '63 CX' (X == number of reset left). 9. The command returns '90 00'. 10. The command returns '63 CX'. Verify that X from this step = X from step 8. 11. The command returns '6A 80' (incorrect parameter in command data field) and the retry counter remains unchanged. 12. The command returns '6A 80' (incorrect parameter in command data field). The retry counter remains unchanged. 13. The command returns: • '63 CX' (X==number of resets left). • '69 83' (reset operation blocked) – when the command is invoked after the value of X becomes zero. NOTE: Testing this condition may leave the card unusable in some implementations for all operations related to the key reference associated with this reset counter.
Postcondition(s)	No further resets of reference data associated with key reference possible.

C.2.3.2 Contactless Interface

Purpose	Validates that the RESET RETRY COUNTER command cannot be issued through the contactless interface without secure messaging.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader.

	4. The value of the counter reference data (PUK) is recorded.
Test Scenario	`Repeat steps 1, 2, and 5 of test C.2.3.1`
Expected Result(s)	1. Step 1 referenced above returns the application property template with the status word '90 00' at the end. 2. Steps 2 and 5 referenced above return an error status word.
Postcondition(s)	Reference data associated with key reference is not changed. Retry counter value associated with the key reference is not reset. The reset counter value is unchanged.

C.2.3.3 Secure Messaging Interface

Purpose	Validates that the RESET RETRY COUNTER command cannot be issued through the secure messaging interface.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.3 2. AS05.01, AS03.07, AS05.30 through AS05.33
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. Secure messaging session keys have been established and secure messaging is used in the test scenario. 5. The value of the counter reference data (PUK) is recorded.

Test Scenario	`1. Repeat step 1 of test C.2.3.1 without secure messaging.`
	`2. Repeat steps 2 and 5 of test C.2.3.1 using the '0C' CLA byte`

Expected Result(s)	1. Step 1 referenced above returns the application property template with the status word '90 00' at the end. 2. Steps 2 and 5 referenced above return an error status word.
Postcondition(s)	Reference data associated with key reference is not changed. Retry counter value associated with the key reference is not reset. The reset counter value is unchanged.

C.2.3.4 Virtual Contact Interface

Purpose	Validates that the RESET RETRY COUNTER command cannot be issued through the VCI.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.3 2. AS05.01, AS03.07, AS05.30 through AS05.33
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader.

	4. There exists a valid VCI connection to the card. 5. The value of the counter reference data (PUK) is recorded.

Test Scenario	`1. Repeat step 1 of test C.2.3.1 without secure messaging.`
	`2. Repeat steps 2 and 5 of test C.2.3.1 using the '0C' CLA byte`

Expected Result(s)	1. Step 1 referenced above returns the application property template with the status word '90 00' at the end. 2. Steps 2 and 5 referenced above return an error status word.
Postcondition(s)	Reference data associated with key reference is not changed. Retry counter value associated with the key reference is not reset. The reset counter value is unchanged.

C.2.4 GENERAL AUTHENTICATE card command

C.2.4.1 Contact Interface

Purpose	Validates the GENERAL AUTHENTICATE command to: 1. Authenticate the PIV Card Application to the Test Toolkit Application (INTERNAL AUTHENTICATE). 2. Authenticate the client application (EXTERNAL AUTHENTICATE). 3. Two-way authentication of PIV Card Application and Test Toolkit Application (MUTUAL AUTHENTICATE). 4. Sign with the '9C' digital signature private key. 5. Enable key-establishment functionality with the '9D' key management private key. 6. Enable key history mechanism functionality with retired key management private keys. 7. Ensure neither the PIV Secure Messaging key nor the associated key-establishment protocol is used inappropriately.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.4 2. AS05.01, AS03.06, AS05.25, AS05.34 through AS05.36B
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. The security status indicator is set to FALSE for all authenticators.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Perform step 2) of 2a in C.2.1.1 to verify cardholder's PIV Card Application PIN.`
	`3. (Internal Authenticate using an asymmetric key) Send GENERAL AUTHENTICATE card command`
	`• CLA is set to:`
	`• '00' if command chaining is not needed or`

        - '10' if command chaining is used. (The last chain of the command sets CLA to '00')
    - P1, algorithm reference, is set to '07' or '11'
    - P2, key reference, is set to '9A' indicating the PIV Authentication key
    - Data field in the command is to include '81' specifying a challenge, followed by a randomly generated challenge, and '82 00' in order to request a response

NOTE: The following test invocation (step 4) is to be performed only if the PIV Card Application supports the symmetric Card Authentication key.

4. (Internal Authenticate using a symmetric key) Send GENERAL AUTHENTICATE card command
    - CLA is set to '00'
    - P1, algorithm reference, is set to '00', '03','08', '0A', or '0C'
    - P2, key reference, is set to '9E' indicating the Card Authentication key
    - Data field in the command is to include '81' specifying a challenge, followed by a randomly generated challenge, and '82 00' in order to request a response

NOTE: The following four test invocations (5a, 5b, 6a, and 6b) are to be performed only if the PIV Card Application supports the use of the '9B' key.

5. (Mutual Authenticate using a symmetric key)
    5a. Send GENERAL AUTHENTICATE card command
        - CLA is set to '00'
        - P1, algorithm reference, is set to '00', '03', '08', '0A', or '0C'
        - P2, key reference, is set to '9B'
        - Data field in the command is to include '80' requesting a witness from the PIV Card Application

    5b. Send GENERAL AUTHENTICATE card command
        - CLA is set to '00'
        - P1, algorithm reference is set to the same value as specified in step 5a
        - P2, key reference is set to '9B'
        - Data field in the command is to include '80' followed by decryption of the encrypted challenge sent by the card application and '81' followed by another challenge and then '82 00'
6. (External Authenticate using a symmetric key)
    6a. Send GENERAL AUTHENTICATE card command
        - CLA is set to '00'
        - P1, algorithm reference, is set to '00', '03', '08', '0A', or '0C'
        - P2, key reference, is set to '9B'
        - Data field in the command is to include '81' followed by '00' indicating it is a request for challenge

    6b. Send GENERAL AUTHENTICATE card command
        - CLA is set to '00'

- P1, algorithm reference, is set to the same value as in step 6a
- P2, key reference, is set to '9B'
- Data field in the command is to include '82' followed by encrypted challenge

7. Test the correct functionality of the digital signature key ('9C'):

   7a. Perform step 2) of 2a in C.2.1.1 to verify cardholder’s PIV Card Application PIN and repeat step 3 with P2 set to '9C', P1 (algorithm reference) set to '07', '11', or '14' and template '81' in the data field containing a hashed message

   7b. Repeat step 3 (without PIN verification). Set P2 to '9C', P1 (algorithm reference) to '07', '11', or '14' and include template '81' in the data field containing a hashed message

8. Repeat step 3 with P2 set to '9D', P1 (algorithm reference) set to '07', '11', or '14' and include template '81' containing an encrypted key (in case of P1 ='07') or template '85' containing the other party’s public key[8] (in case of P1 = '11' or '14')
9. Repeat step 3 with P2 set to '9E' (Card Authentication key) and P1 (algorithm reference) set to '07' or '11' and the template '81' containing a randomly generated challenge
10. If the Key History Object is supported:

    Send GET DATA command with

    - Data field of the command containing the tag of the Key History Object data object. Retrieve the key history’s data elements:
    - If keysWithOnCardCerts = 0 and keysWithOffCardCerts > 0
      - o Read the certificate(s) and key references (pairs) from the vendor provided URL file. For each key reference value in the range (0x95 – keysWithOffCardCerts + 1) through 0x95, verify that the provided URL file includes that key reference, issue a challenge for that key reference, and verify the response using the public key from the corresponding certificate from the provided URL file
    - If keysWithOnCardCerts > 0 and keyWithOffCardCerts = 0
      - o For each key reference value in the range 0x82 through (0x82 + keysWithOnCardCerts – 1), read the certificates from the card. Issue a challenge for each retired private key,[9] and verify the response using the public key from the corresponding certificate

[8] Template '85' contains the other party’s public key, a point on Curve P-256 or P-384, encoded as '04' || X || Y, without the use of point compression, as described in Section 2.3.3 of [SEC1].

[9] See Table 7 of SP 800-73-4 Part 1 for the association of certificate BER-TLV tags to corresponding key reference values.

	• If keysWithOnCardCerts > 0 and keyWithOffCardCerts > 0 o For each key reference value in the range 0x82 through (0x82 + keysWithOnCardCerts – 1) and in the range (0x95 – keysWithOffCardCerts + 1) through 0x95, verify that the provided URL file includes that key reference, issue a challenge for that key reference, and verify the response using the public key from the corresponding certificate from the provided URL file 11. Repeat step 3 with an invalid value of algorithm reference (P1) and/or key reference (P2) 12. Repeat step 3 with an invalid value in data field (improper challenge length for the chosen algorithm) 13. Reset the security status indicator of the PIV Card Application PIN by performing VERIFY with a wrong PIN 14. If the application property template obtained in step 1 indicates that the Global PIN satisfies the PIV ACRs for command execution and data access, then perform step 2) of 2b in C.2.1.1 to verify cardholder’s Global PIN and repeat steps 3, 7-10, and 13 (but performing the VERIFY using the Global PIN in steps 7a and 13). 15. If the application property template obtained in step 1 indicates that OCC satisfies the PIV ACRs for command execution and data access, then • Perform step 1) of 2c in C.2.1.1 to verify cardholder’s OCC and repeat steps 3, 7-10, and 13 (but performing the VERIFY using key reference '96' in steps 7a and 13). • Perform step 4) of 2c in C.2.1.1 to verify cardholder’s OCC and repeat steps 3, 7-10, and 13 (but performing the VERIFY using key reference '97' in steps 7a and 13). 16. Repeat steps 3, 7b, 8, and 9 17. Repeat steps 4, 6, and 10, if the key types specified in the tests are supported 18. Send GENERAL AUTHENTICATE card command • P1, algorithm reference, is set to '27' or '2E', as indicated by the 0xAC tag obtained from the application property template in step 1 • P2, key reference, is set to '9A' indicating the PIV Authentication key 19. Repeat step 18 with P2, key reference, values of '00', '80', '81', '98', '9B', '9C', '9D', '9E', and all retired key management keys 20. Send GENERAL AUTHENTICATE card command • P1, algorithm reference, is set to '11' or '14' • P2, key reference, is set to '04' indicating the PIV Secure Messaging key NOTE: The following test invocation (step 21) is only performed if the PIV Card Application supports the use of the '04' key 21. Send GENERAL AUTHENTICATE card command

	• `P1, algorithm reference, is set to '27' or '2E', as indicated by the 0xAC tag obtained from the application property template in step 1` • `P2, key reference, is set to '04' indicating the PIV Secure Messaging key`
Expected Result(s)	1. Command returns the application property template with the status word '90 00' at the end. 2. The command returns '90 00' 3. The command returns the signed challenge with '90 00' at the end. Verify the signed challenge. 4. The command returns the encrypted challenge with '90 00' at the end. Decrypt the encrypted challenge and compare it to the one sent to the card. 5a. The PIV Card Application returns with the encryption of a challenge followed by '90 00'. 5b. The PIV Card Application verifies the witness and then responds with encryption of the challenge sent by Test Toolkit Application followed by '90 00'. Decrypt the encrypted challenge and compare it to the one sent to the card. 6a. The PIV Card Application returns a challenge followed by '90 00'. 6b. The Test Toolkit Application responds with encryption of the challenge sent by PIV Card Application and the card returns '90 00'. 7a. The command returns the signed data with '90 00' at the end. Verify the signature using the public key from the digital signature certificate and the hash sent to the card. 7b. The command returns '69 82' (security status not satisfied). 8. For algorithm reference '07' as P1 value, the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. For algorithm reference '11' or '14' as P1 value, the command returns the shared secret Z [10] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card. 9. The command returns the signed challenge with '90 00' at the end. Verify the signed challenge. 10. The GET DATA commands return the requested data along with '90 00'. Each GENERAL AUTHENTICATE command returns either 1) the transported key with '90 00' at the end or 2) the shared secret Z with '90 00' at the end. • For key transport (as indicated by algorithm reference '06' or '07' as P1 value), the command returns the transported key

[10] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2

	with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. • For ECDH, (as indicated by algorithm reference '11' or '14' as P1 value), the command returns the shared secret Z [11] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card. 11. The command returns '6A 86' (incorrect parameter in P1 or P2). 12. The command returns '6A 80' (incorrect parameter in command data field). 13. The security state is reset. 14. • Repeated step 3: The command returns the signed challenge with '90 00' at the end. Verify the signed challenge. • Repeated step 7a: The command returns the signed data with '90 00' at the end. Verify the signature using the public key from the digital signature certificate and the hash sent to the card. • Repeated step 7b: The command returns '69 82' (security status not satisfied). • Repeated step 8: For algorithm reference '07' as P1 value, the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. For algorithm reference '11' or '14' as P1 value, the command returns the shared secret Z [12] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card. • Repeated step 9: The command returns the signed challenge with '90 00' at the end. Verify the signed challenge. • Repeated step 10: The GET DATA commands return the requested data along with '90 00'. Each GENERAL AUTHENTICATE command returns either 1) the transported key with '90 00' at the end or 2) the shared secret Z with '90 00' at the end. o For key transport (as indicated by algorithm reference '06' or '07' as P1 value), the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. o For ECDH, (as indicated by algorithm reference '11' or '14' as P1 value), the command returns the shared

[11] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2
[12] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2

secret Z [13] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card.

- Repeated step 13: The security state is reset.

15.

- Repeated step 3: The command returns the signed challenge with '90 00' at the end. Verify the signed challenge.
- Repeated step 7a: The command returns the signed data with '90 00' at the end. Verify the signature using the public key from the digital signature certificate and the hash sent to the card.
- Repeated step 7b: The command returns '69 82' (security status not satisfied).
- Repeated step 8: For algorithm reference '07' as P1 value, the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. For algorithm reference '11' or '14' as P1 value, the command returns the shared secret Z [14] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card.
- Repeated step 9: The command returns the signed challenge with '90 00' at the end. Verify the signed challenge.
- Repeated step 10: The GET DATA commands return the requested data along with '90 00'. Each GENERAL AUTHENTICATE command returns either 1) the transported key with '90 00' at the end or 2) the shared secret Z with '90 00' at the end.
    - For key transport (as indicated by algorithm reference '06' or '07' as P1 value), the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card.
    - For ECDH, (as indicated by algorithm reference '11' or '14' as P1 value), the command returns the shared secret Z [15] with '90 00' at the end. Compare the shared secret computed by the card with the shared secret computed off card.
- Repeated step 13: The security state is reset.

16. For the referenced steps 3, 7b, and 8, the command returns '69 82' (security status not satisfied). For the referenced step 9, command

[13] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2
[14] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2
[15] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2

	returns the signed challenge with '90 00' at the end. Verify the signed challenge. 17. The command returns: • For referenced step 4, the command returns the encrypted challenge with '90 00' at the end. Decrypt the encrypted challenge and compare it to the one sent to the card. • Referenced step 6a: The PIV Card Application returns a challenge followed by '90 00'. • Referenced step 6b. The Test Toolkit Application responds with encryption of the challenge sent by PIV Card application and the card returns '90 00' • For referenced step 10, the GET DATA commands return '90 00' and the requested data objects. The GENERAL AUTHENTICATE commands return '69 82' (security status not satisfied) NOTE: On Steps 3, 7a, 9, 14, 15, and 16: If ECDSA with algorithm '11' (in case of '9A', '9C', or '9E') or '14' (in case of '9C') is used, the response data field contains r and s.[16] 18. The command returns '6A 86' (incorrect parameter in P1 or P2). 19. The commands return '6A 86' (incorrect parameter in P1 or P2). 20. The command returns '6A 86' (incorrect parameter in P1 or P2). 21. The command returns '90 00' – secure messaging session keys are established. Use the information returned by the PIV Card Application to derive the session keys and verify the key confirmation AuthCryptogram$_{ICC}$.
Postcondition(s)	N/A

C.2.4.2 Contactless Interface

Purpose	Validates internal authentication and mutual authentication of the PIV Card and the Test Toolkit to ensure that the private keys in use are accessible only through the appropriate interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader.

[16] r and s are DER encoded with the following ASN.1 structure:

```
Ecdsa-Sig-Value ::= SEQUENCE {
        r    INTEGER,
        s    INTEGER  }
```

Test Scenario	`1. Send SELECT card command with`
	• `AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Repeat steps 3, 7b, 8, and 9 of C.2.4.1`
	`3. Repeat steps 4 and 6 of C.2.4.1, if the key types specified in the tests are supported`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. Referenced Steps: • Steps 3, 7b, and 8 referenced in C.2.4.1 the command returns '69 82' (security status not satisfied). • Step 9 referenced in C.2.4.1 returns the signed challenge with '90 00' at the end. Verify the signed challenge. 3. Referenced Steps: • Step 4 referenced in C.2.4.1 the command returns the encrypted challenge with '90 00' at the end. Decrypt the encrypted challenge and compare it to the one sent to the card. • Step 6 referenced in C.2.4.1 returns '69 82' (security status not satisfied). NOTE: For step 9: If ECDSA with algorithm '11' is used, the response data field contains r and s.
Postcondition(s)	N/A

C.2.4.3 Secure Messaging Interface

Purpose	Validates internal authentication and mutual authentication of the PIV Card and the Test Toolkit to ensure that the private keys in use are accessible only through the appropriate interface.
Reference(s)	3. SP 800-73-4 Part 2, Table 2 4. AS05.03, AS05.34, AS05.36A, AS05.36B
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. Secure messaging session keys have been established and secure messaging is used in the test scenario.

Test Scenario	`1. Send SELECT card command with`
	• `AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Repeat steps 2 and 3 from C.2.4.2 using the '0C' CLA byte`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The referenced commands return the same expected results as in C.2.4.2.

Postcondition(s)	N/A

C.2.4.4 Virtual Contact Interface

Purpose	1. Validates the GENERAL AUTHENTICATE command to: 2. Authenticate the PIV Card Application to the Test Toolkit Application (INTERNAL AUTHENTICATE). 3. Authenticate the client application (EXTERNAL AUTHENTICATE). 4. Two-way authentication of PIV Card Application and Test Toolkit Application (MUTUAL AUTHENTICATE). 5. Sign with the '9C' digital signature private key. 6. Enable key-establishment functionality with the '9D' key management private key. 7. Enable key history mechanism functionality with retired key management private keys.
Reference(s)	1. SP 800-73-4 Part 2, Section 3.2.4 2. AS05.01, AS03.06, AS05.25, AS05.34 through AS05.36B
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. The PIV Card Application is the currently selected application on the card. 5. The security status indicator is set to FALSE for all authenticators except the pairing code. 6. There exists a valid VCI connection to the card.

Test Scenario	`1. Send SELECT card command without secure messaging with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Perform steps 2 – 17 in C.2.4.1 using the '0C' CLA byte`

Expected Result(s)	1. Command returns the application property template with the status word '90 00' at the end. 2. See expected results for C.2.4.1 except for steps 5 and 6, which will result in '69 82' (security status not satisfied).
Postcondition(s)	N/A

C.3 Card Commands for Credential Initialization and Administration

C.3.1 PUT DATA Command

C.3.1.1 Contact Interface

Purpose	Validates that the PUT DATA command exhibits the appropriate behavior under the following conditions: 1. Without the security condition is satisfied. 2. After the security condition is satisfied.
Reference(s)	1. SP 800-73-4, Part 2, Section 3.3.1 2. AS05.01, AS05.02, AS05.37
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. The PIV Card Application is the currently selected application on the card. 4. The mutual authentication of PIV Card Application and the Test Toolkit Application has not been performed.
Test Scenario	1. Send PUT DATA card command with • CLA is set to: • '00' if command chaining is not needed or • '10' if command chaining is used. (The last chain of the command sets CLA to '00') • Data field in the command is to include the tag of the Card Capability Container data object • Data field in the command is to include the data that will replace the Card Capability Container 2. Repeat step 1 with • Data field in the command is to include the tag of the CHUID data object • Data field in the command is to include the data content that will replace the CHUID 3. Repeat step 1 with • Data field in the command is to include the tag of the X.509 Certificate for PIV Authentication data object • Data field in the command is to include data content that will replace the X.509 Certificate for PIV Authentication 4. Repeat step 1 with • Data field in the command is to include the tag of the Cardholder Fingerprints data object • Data field in the command is to include data content that will replace the Cardholder Fingerprints 5. If the card supports the Printed Information data object, repeat step 1 with • CLA='00' • Data field in the command is to include the tag of the Printed Information data object

	• Data field in the command is to include the data content that will replace the Printed Information 6. Repeat step 1 with • Data field in the command is to include the tag of the Cardholder Facial Image data object • Data field in the command is to include the data content that will replace the Cardholder Facial Image 7. Repeat step 1 with • Data field in the command is to include the tag of the X.509 Certificate for Digital Signature data object • Data field in the command is to include the data content that will replace the X.509 Certificate for Digital Signature 8. Repeat step 1 with • Data field in the command is to include the tag of the X.509 Certificate for Key Management data object • Data field in the command is to include the data content that will replace the X.509 Certificate for Key Management 9. Repeat step 1 with • Data field in the command is to include the tag of the X.509 Certificate for Card Authentication data object • Data field in the command is to include the data content that will replace the X.509 Certificate for Card Authentication 10. If the card supports the Discovery Object, repeat step 1 with • CLA = '00' • Data field in the command is to include the tag of the Discovery Object • Data field in the command is to include the data content that will replace the Discovery Object 11. Repeat step 1 with • Data field in the command is to include the tag of the Security Object • Data field in the command is to include the data content that will replace the Security Object 12. If the card supports the Key History Object, repeat step 1 with • CLA = '00' • Data field in the command is to include the tag of the Key History object • Data field in the command is to include the data content that will replace the Key History Object 13. If the card supports Key History Object, repeat step 1 for each implemented Retired X.509 Certificate for Key Management with • Data field in the command is to include the tag of one of the 20 Retired X.509 Certificates for Key Management • Data field in the command is to include the data content that will replace the Retired X.509 Certificate for Key Management

	14. If the card supports the Cardholder Iris Images data object, repeat step 1 with • Data field in the command is to include the tag of the Cardholder Iris Images data object • Data field in the command is to include the data content that will replace the Cardholder Iris Images data object 15. If the card supports secure messaging for non-card-management operations, repeat step 1 with • Data field in the command is to include the tag of the Secure Messaging Certificate Signer data object • Data field in the command is to include the data content that will replace the Secure Messaging Certificate Signer object 16. If the card supports the virtual contact interface, repeat step 1 with • CLA = '00' • Data field in the command is to include the tag of the Pairing Code Reference Data Container data object • Data field in the command is to include the data content that will replace the Code Reference Data Container data object 17. If the card supports OCC, repeat step 1 with • CLA = '00' • Data field in the command is to include the tag of the Biometric Information Templates Group Template object • Data field in the command is to include the data content that will replace the Biometric Information Templates Group Template data object NOTE: The following tests are to be performed only if the PIV Card Application supports the use of the '9B' key 18. Perform mutual authentication of PIV Card Application and the Test Toolkit Application using steps 5a and 5b of C.2.4.1 (GENERAL AUTHENTICATE) 19. Repeat steps 1-16 with GET DATA command immediately following each PUT DATA and verifying whether the same data that is input with PUT DATA command is returned by GET DATA command
Expected Result(s)	1. In steps 1 through 16, commands return '69 82' (security status not satisfied). 2. The two test invocations referred to in step 18 should return the same responses as 5a and 5b of Expected Results under test C.2.4.1. 3. In step 19, all commands return '90 00', and input and output data strings match.
Postcondition(s)	The contents of each object have been overwritten with the new values provided in step 19.

C.3.1.2 Contactless Interface

Purpose	Validates that the PUT DATA command cannot be issued through the contactless interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing values of all data objects have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader. 4. No other contactless card is within the proximity of the reader.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Repeat steps 1-17 of C.3.1.1`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The commands return an error status word for referenced steps 1-17.
Postcondition(s)	The data container values remain unchanged.

C.3.1.3 Secure Messaging Interface

Purpose	Validates that the PUT DATA command cannot be issued through the secure messaging interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing values of all data objects have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader. 4. No other contactless card is within the proximity of the reader. 5. Secure messaging session keys have been established and secure messaging is used in the test scenario.

Test Scenario	`1. Send SELECT card command without secure messaging with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Repeat steps 1-17 of C.3.1.1 using the '0C' CLA byte`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. The commands return an error status word for referenced steps 1-17.
Postcondition(s)	The data container values remain unchanged.

C.3.1.4 Virtual Contact Interface

Purpose	Validates that the PUT DATA command cannot be issued through the VCI.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing values of all data objects have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader. 4. No other contactless card is within the proximity of the reader. 5. The PIV Card Application is the currently selected application on the card. 6. There exists a valid VCI connection to the card.
Test Scenario	`Repeat steps 1-17 of C.3.1.1 using the '0C' CLA byte`
Expected Result(s)	The commands return an error status word for referenced steps 1-17.
Postcondition(s)	The data container values remain unchanged.

C.3.2 GENERATE ASYMMETRIC KEY PAIR command

C.3.2.1 Contact Interface

Purpose	Validates that the card executes the GENERATE ASYMMETRIC KEY PAIR command for the following conditions: 1. Without the security condition satisfied. 2. After the security condition (authenticating with the PIV Card Application Administrator) is satisfied.
Reference(s)	1. SP 800-73-4, Section Part 2, 3.3.2 2. AS05.01, AS05.38 through AS05.40
Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. The PIV Card Application is the currently selected application on the card. 4. The mutual authentication of PIV Card Application and the Test Toolkit Application has not been performed.

Test Scenario	`1. Send GENERATE ASYMMETRIC KEY PAIR card command with`
	• `P2 is set to value '9A'`
	• `Data field in the command is to include either '07' or '11' as the cryptographic mechanism identifier`
	`2. Send GENERATE ASYMMETRIC KEY PAIR card command with`
	• `P2 is set to value '9C'`
	• `Data field in the command is to include either '07', '11', '14' as the cryptographic mechanism identifier`

3. If the PIV Card Application supports on-card generation of the key management key send GENERATE ASYMMETRIC KEY PAIR card command with
   - P2 is set to value '9D'
   - Data field in the command is to include either '07', '11', '14' as the cryptographic mechanism identifier
4. If the PIV Card Application supports on-card generation of the asymmetric Card Authentication key send GENERATE ASYMMETRIC KEY PAIR card command with
   - P2 is set to value '9E'
   - Data field in the command is to include either '07' or '11' as the cryptographic mechanism identifier
5. If the card supports secure messaging send GENERATE ASYMMETRIC KEY PAIR card command with
   - P2 is set to value '04'
   - Data field in the command is to include either '11' or '14' as the cryptographic mechanism identifier

NOTE: The following tests are to be performed only if the PIV Card Application supports the use of the key '9B'.

6. Perform mutual authentication of PIV Card Application and the Test Toolkit Application using steps 5a and 5b of C.2.4.1 (GENERAL AUTHENTICATE)
7. Repeat steps 1-5
8. Send VERIFY card command with
   - P2, key reference value is set to '80'
   - Data field of the command will contain the correct PIV Card Application PIN value, padded with 'FF' to complete the total length of the value to 8 bytes
9. Send GENERAL AUTHENTICATE card command with
   - CLA is set to:
     - '00' if command chaining is not needed or
     - '10' if command chaining is used. (The last chain of the command sets CLA to '00')
   - P1, algorithm reference, is set to '07' or '11'
   - P2, key reference, is set to '9A' indicating the PIV Authentication key
   - Data field in the command is to include '81' specifying a challenge, followed by a randomly generated challenge, and '82 00' in order to request a response
10. Repeat step 8 to verify cardholder’s PIV Card Application PIN and repeat step 9 with P2 set to '9C', P1 (algorithm reference) set to '07', '11', or '14' and template '81' in the data field containing a hashed message
11. Repeat step 9 with P2 set to '9D', P1 (algorithm reference) set to '07', '11', or '14' and include template '81' containing a key encrypted using the key management public key returned in step 7 (in case of P1 ='07') or template '85' containing the other party’s public key[17] (in case of P1 = '11' or '14')

[17] Template '85' contains the other party’s public key, a point on Curve P-256 or P-384, encoded as '04' || X || Y, without the use of point compression, as described in Section 2.3.3 of [SEC1].

	12. Repeat step 9 with P2 set to '9E' (Card Authentication key) and P1 (algorithm reference) set to '07' or '11' and the template '81' containing a randomly generated challenge 13. Send GENERAL AUTHENTICATE card command • P1, algorithm reference, is set to '27' or '2E' • P2, key reference, is set to '04' indicating the PIV Secure Messaging key 14. Repeat step 1 with the cryptographic mechanism identifier value in the data field set to a value that is not supported by the card. 15. Repeat step 1 with P2 set to a key reference value that is not supported by the card
Expected Result(s)	1. Command returns '69 82' (security status not satisfied). 2. Command returns '69 82' (security status not satisfied). 3. Command returns '69 82' (security status not satisfied). 4. Command returns '69 82' (security status not satisfied). 5. Command returns '69 82' (security status not satisfied). 6. The two test invocations referred to in step 2 should return the same responses as 5a and 5b of Expected Results under test C.2.4.1. 7. For referenced steps 1 through 5, command returns status word '90 00' and the data field contains the '7F49' template with the generated public key, which consists of either a modulus and public exponent (RSA) or a point (elliptic curve cryptography). 8. The command returns '90 00' 9. The command returns the signed challenge with '90 00' at the end. Verify the signed challenge using the PIV Authentication public key that was returned in step 7. 10. The VERIFY command returns '90 00' The GENERAL AUTHENTICATE command returns the signed data with '90 00' at the end. Verify the signature using the hash sent to the card and the digital signature public key that was returned in step 7. 11. For algorithm reference '07' as P1 value, the command returns the transported key with '90 00' at the end. Compare the plaintext key to the one received in the response from the card. For algorithm reference '11' or '14' as P1 value, the command returns the shared secret Z [18] with '90 00' at the end. Compare the shared secret computed by the card with shared secret computed off the card (using the key management public key that was returned in step 7). 12. The command returns the signed challenge with '90 00' at the end. Verify the signed challenge using the Card Authentication public key that was returned in step 7.

[18] Z is the X coordinate of point P as defined in SP 800-56A, Section 5.7.1.2

	13. The command returns '90 00' – secure messaging session keys are established. Use the information returned by the PIV Card Application to derive the session keys and verify the key confirmation AuthCryptogram$_{ICC}$ (but use the public key that was returned in step 7 to compute the ECC_CDH primitive). 14. The command returns '6A 80' (incorrect parameter command data field). 15. The command returns '6A 86' (incorrect parameter P2).
Postcondition(s)	The on card private keys have changed to the new computed value.

C.3.2.2 Contactless Interface

Purpose	Validates that the GENERATE ASYMMETRIC KEY PAIR command cannot be issued through the contactless interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing contents of the public key data object have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader. 4. No other contactless card is within the proximity of the reader.

Test Scenario	`1. Send SELECT card command with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Perform steps 1-5 of test C.3.2.1`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. Referenced steps 1-5 from C.3.2.1 return an error status word for all key references.
Postcondition(s)	N/A

C.3.2.3 Secure Messaging Interface

Purpose	Validates that the GENERATE ASYMMETRIC KEY PAIR command cannot be issued through the secure messaging interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing contents of the public key data object have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader.

	4. No other contactless card is within the proximity of the reader. 5. Secure messaging session keys have been established and secure messaging is used in the test scenario.

Test Scenario	`1. Send SELECT card command without secure messaging with`
	`• AID == 'A0 00 00 03 08 00 00 10 00 01 00'`
	`2. Perform steps 1-5 of test C.3.2.1 using the '0C' CLA byte`

Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. Referenced steps 1-5 from C.3.2.1 return an error status word for all key references.
Postcondition(s)	N/A

C.3.2.4 Virtual Contact Interface

Purpose	Validates that the GENERATE ASYMMETRIC KEY PAIR command cannot be issued through the VCI.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.03
Precondition(s)	1. The existing contents of the public key data object have been recorded. 2. The IUT is placed within the reading range of the contactless reader. 3. There exists a valid PC/SC connection between the test system and the contactless reader. 4. No other contactless card is within the proximity of the reader. 5. The PIV Card Application is the currently selected application on the card. 6. There exists a valid VCI connection to the card.
Test Scenario	`Perform steps 1-5 of test C.3.2.1 using the '0C' CLA byte`
Expected Result(s)	Referenced steps 1-5 from C.3.2.1 return an error status word for all key references.
Postcondition(s)	N/A

C.3.3 Secure Messaging Error Handling

The following tests are applicable to all cards that support secure messaging as specified in NIST SP 800-73-4 Part 2 Section 4.

C.3.3.1 Contact Interface

Purpose	Validates that the card handles secure messaging error conditions properly.
Reference(s)	1. SP 800-73-4, Section Part 2, 4.2.7 2. AS05.43-R4

Precondition(s)	1. The IUT is inserted into the contact reader. 2. There exists a valid PC/SC connection between the test system and the contact reader. 3. Secure messaging session keys have not been established.
Test Scenario	1. Send SELECT card command with • AID == 'A0 00 00 03 08 00 00 10 00 01 00' 2. Send GET DATA command with • CLA is set to a value of '0C' • The BER-TLV encoded encrypted PIV data field shall be formatted as follows: '87 11 01 (16 bytes of random data to simulate one block of encrypted data)' • The BER-TLV encoded C-MAC shall be formatted as follows: '8E 08 (8 bytes of random data to simulate MAC value)' 3. Send GENERAL AUTHENTICATE card command • P1, algorithm reference, is set to '27' or '2E', as indicated by the 0xAC tag obtained from the application property template in step 1 • P2, key reference, is set to '04' indicating the PIV Secure Messaging key 4. Send GET DATA command using the '0C' CLA byte with • The encrypted data field of the command containing the tag of the CHUID data object • The command is properly formatted with the exception of the required BER-TLV encoded C-MAC, which shall be absent 5. Send GENERAL AUTHENTICATE card command • P1, algorithm reference, is set to '27' or '2E', as indicated by the 0xAC tag obtained from the application property template in step 1 • P2, key reference, is set to '04' indicating the PIV Secure Messaging key 6. Send GET DATA command with • The encrypted data field of the command containing the tag of the CHUID data object • The command is properly formatted however the required BER-TLV encoded C-MAC is incorrect
Expected Result(s)	1. The command returns the application property template with the status word '90 00' at the end. 2. Command returns '69 82' (security status not satisfied). 3. Command returns '90 00' (successful execution). 4. Command returns '69 87' (expected secure messaging data objects are missing). 5. Command returns '90 00' (successful execution). 6. Command returns '69 88' (secure messaging data objects are incorrect).
Postcondition(s)	N/A

C.3.3.2 Contactless Interface

Purpose	Validates that the card handles secure messaging error conditions properly when using the contactless interface.
Reference(s)	1. SP 800-73-4 Part 2, Table 2 2. AS05.43-R4
Precondition(s)	1. The IUT is placed within the reading range of the contactless reader. 2. There exists a valid PC/SC connection between the test system and the contactless reader. 3. No other contactless card is within the proximity of the reader. 4. Secure messaging keys have not been established.
Test Scenario	`Perform the same steps as in C.3.3.1`
Expected Result(s)	The commands will have the same expected results as C.3.3.1
Postcondition(s)	N/A

Appendix D—Acronyms

The following acronyms and abbreviations are used throughout this publication:

AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
BER-TLV	Basic Encoding Rules Tag-Length-Value
CHUID	Card Holder Unique Identifier
DTR	Derived Test Requirement
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
HSPD	Homeland Security Presidential Directive
ICC	Integrated Circuit Chip
IEC	International Electrotechnical Commission
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITL	Information Technology Laboratory
IUT	Implementation Under Test
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
PC	Personal Computer
PIN	Personal Identification Number
PIV	Personal Identity Verification
PIX	Proprietary Identifier eXtension
PUK	PIN Unblocking Key
RID	Registered application provider IDentifier
SM	Secure Messaging
SP	Special Publication
TRD	Test Run Detail
TRS	Test Results Summary
VCI	Virtual Contact Interface

Appendix E—References

[FIPS 201]	Federal Information Processing Standard 201-2, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, August 2013. http://dx.doi.org/10.6028/NIST.FIPS.201-2
[HSPD 12]	Homeland Security Presidential Directive-12, *Policies for a Common Identification Standard for Federal Employees and Contractors*, August 27, 2004. http://www.dhs.gov/homeland-security-presidential-directive-12
[ISO/IEC 7816]	ISO/IEC 7816 (Parts 4, 5, 6, 8, and 9), *Information technology — Identification cards — Integrated circuit(s) cards with contacts*.
[ISO/IEC 14443]	ISO/IEC14443 (Parts 1,2,3,4), *Identification cards – Contactless integrated circuit(s) cards – Proximity cards*.
[SEC1]	Standards for Efficient Cryptography Group (SECG), "SEC 1: Elliptic Curve Cryptography," Version 1.0, September 2000.
[SP 800-56A]	NIST Special Publication 800-56A Revision 2, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, May 2013. http://dx.doi.org/10.6028/NIST.SP.800-56Ar2
[SP 800-73]	NIST Special Publication 800-73-4, *Interfaces for Personal Identity Verification*, May 2015 (updated February 8, 2016). http://dx.doi.org/10.6028/NIST.SP.800-73-4
[SP 800-76]	NIST Special Publication 800-76-2, *Biometric Specifications for Personal Identity Verification*, July 2013. http://dx.doi.org/10.6028/NIST.SP.800-76-2
[SP 800-78]	NIST Special Publication 800-78-4, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification*, May 2015. http://dx.doi.org/10.6028/NIST.SP.800-78-4
[SP 800-85B]	Draft NIST Special Publication 800-85B-4, *PIV Data Model Test Guidelines*, August 2014. http://csrc.nist.gov/publications/drafts/800-85B/sp800_85b-4_draft.pdf