FIPS 140-2
SECURITY POLICY

MODEL 400 SMART CARD

Version 1.1

SafeNet
The Foundation of Information Security

4690 Millenium Drive
Belcamp, MD 21017

Table of Contents

1. Introduction

1.1. Scope

This document describes the cryptographic module security policy for the SafeNet Model 400 smart card (Hardware version: 2.0, Firmware Version 3.0). It contains specification of the security rules, under which the cryptographic module operates, including the security rules derived from the requirements of the FIPS 140-2 standard and the additional security rules imposed by the Biometric authentication and FIPS 201 application executable (PIV EXF) Version 19 loaded on the 400 smart card.

1.2. Overview

The 400 cryptographic module is a dual interface smart card compliant with parts 1 through 4 of the ISO/IEC7816 standard and ISO/IEC14443 Parts 1-4, which define the physical characteristics, contact arrangement /location, electrical characteristics, and interface protocol.

The chip platform operating system is based on SafeNet Crypto Card Operating System (SCCOS version 3.0) and the application extension PIV EXF. Together they manage all the low level resources, cryptographic algorithms implementation, object access control and applications life cycle.

The PIV EXF is a state of the art application designed and developed using SafeNet's extensive experience in secure operating system development. When downloaded to the chip the PIV EXF provides the following high-level services:

- Card command interface for use of the Personal Identity Verification (PIV) card as specified in Special Publication 800-73-1.
- Interoperability across deployments or agencies.
- Authentication of the cardholder and the security officer.

1.3. Model 400 smart card architecture

- The architecture of the Model 400 smart card is different from the FIPS 140-2 Level 2 certified Model 330G3 smart card. SafeNet designed the Model 400 to support Identity-based authentication of the card users and security officers using PIN, Biometrics, and AES, TRIPLE-DES or RSA keys.
- It provides dual interface capabilities needed to comply with the FIPS 201 standard and a secure multi-application environment.

1.4. Related Standards and Documents

CC	ISO 15408 – Information Technology – Security Techniques – Evaluation Criteria for IT Security (Hereafter referred to as Common Criteria or CC)
ISO 7816-1	ISO/IEC 7816-1 (1987): "Identification cards – Integrated circuit(s) cards with contacts, Part 1: Physical characteristics".
ISO 7816-2	ISO/IEC 7816-2 (1988): "Identification cards – Integrated circuit(s) cards with contacts, Part 2: Dimensions and locations of the contacts".
ISO 7816-3	ISO/IEC 7816-3 (1989): "Identification cards – Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols".
ISO 7816-4	ISO/IEC 7816-4: "Identification cards – Integrated circuit(s) cards with contacts, Part 4:
ISO 14443-1	Contactless integrated circuit(s) cards – Proximity cards —Part 1: Physical characteristicss
ISO 14443-2	Contactless integrated circuit(s) cards – Proximity cards — Part 2: Radio frequency power and signal interface
ISO 14443-3	Contactless integrated circuit(s) cards – Proximity cards — Part 3: Initialization and anti-collision
ISO 14443-4	Contactless integrated circuit(s) cards – Proximity cards — Part 4: Transmission protocol
PKCS 1	PKCS #1: RSA Encryption Standard, Version 1.5, November 1993

2. Glossary

Authentication Data	Comprise the officer identifier, certificate, role and privileges.
Bond-out chips	Raw ICs, which have been mounted on a small board. Wire bonds are connected from the IC's input/output pads to the carrier, which has contacts on its reverse side. Bond-out chips are sometimes referred to as a module.
Card disablement	The IC function related to terminating all operations other than possibly some limited audit functions. Card disablement is permanent.
Card embedder	A manufacturer who assembles a card and integrated circuit.
Card holder	A person to whom a card has been legitimately issued (a user).
Card issuer	An institution, which issues cards to cardholders.
Card Operating System (COS)	Operating system developer specific code, written in the microprocessor's native or machine code.
Card reader	A machine capable of reading and/or writing to a card, such as magnetic stripe card or smart card.
Carrier	The holder in which an operational integrated circuit is placed. This is typically the thin, credit card sized piece of plastic that is known as a smart card.
Die	The semiconductor IC without any packaging or connections.
EEPROM	Electrically Erasable Programmable Read Only Memory. A non-volatile memory technology where data can be electrically erased and rewritten.
Failure analysis	The compilation of techniques used by semiconductor development and testing labs to identify the operating problems in newly designed or modified integrated circuits. Such techniques include not only observation (to determine what is not functioning properly) but also modification of IC internal structure (to determine fixes).
First use indication	The IC function related to setting a specific audit bit indicating that the smart card is now in the issued, operational state and can be used for its intended function.
I&A	Identification and Authentication
IC	Integrated Circuit. Electronic component(s) contained on a single chip and designed to perform processing and/or memory functions.
ICC	Integrated Circuit Card. A card into which has been inserted one or more ICs.

ID	Identity (also, a token asserting an identity)
Initialization	The process of writing specific information into Non-Volatile Memory during the early card life cycle.
IP	Internet Protocol
ISO	International Standards Organization
Life cycle identifiers	The specific identification of chip fabricator identifier, operating software identifier, chip module identifier, chip embedder identifier, initialiser identifier, initialization equipment identifier, personaliser identifier and personalization equipment identifier.
Modules	A functional assembly for use with other assemblies. These may be separate parts of an IC (CPU, Coprocessor, ROM, RAM, etc.), bond-out chips, or software components.
NIST	National Institute of Standards and Technologies
Non-volatile memory	A semiconductor memory that retains its content when power is removed. (i.e. ROM, EEPROM, FLASH).
Personalization	The process of writing specific information into the non-volatile memory in preparing the IC for issuance to users.
PIN	Personal Identification Number
Platform	A term representing an operational smart card system.
Post-issuance	The time period during which the smart card is in the hands of the cardholder. In some smart cards, additional functionality can be loaded into the smart card post-issuance.
PP	Protection Profile
RAM	Random Access Memory. A volatile, randomly accessible memory (used in the IC) that requires power to maintain data.
ROM	Read Only Memory. A non-volatile memory (used in the IC) that requires no power to maintain. ROM data is often contained in one of the numerous masks used during manufacture.
RSA	Rivest, Shamir, Adleman (encryption algorithm)
Security Officer	The administrator of the CM system. The security officer has in addition to the administrative privileges also all the privileges a registration officer can have
SHA	Secure Hash Algorithm

Smart card	A shaped piece of plastic or other carrier with a small computer chip embedded into it.
Terminal	The device used in conjunction with the CAD at the point of transaction.

3. Security Levels

The SafeNet Model 400 meets all requirements for FIPS 140-2 level 2. Refer to the following table for individual security requirements:

Security Requirements	Certification Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services, and Authentication	3
Finite State Model	2
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	2
Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)	3
Self Tests	2
Design Assurance	3
Mitigation of Other Attacks	N/A

4. Cryptographic Module Specification

4.1. Cryptographic Boundary

The cryptographic boundary for the SafeNet Model 400 Smart Card is the physical boundary of the card itself. Although the boundary could be defined as the physical micro-module containing the processor chip, the embedding of the micro-module within the plastic card provides no further cryptographic function. The card provides utility to the micro-module as a standalone cryptographic module component and enhances the tamper evidence of the cryptographic module.

Although the Model 400 card and its operating system SCCOS are designed to meet and exceed requirements for FIPS 140-2 level 3, the cryptographic module submitted for testing is a single-chip module combining the SCCOS v3.0 operating system in ROM and the PIV EXF in EEPROM, together to form the cryptographic module that meets the FIPS 140-2 level 2 requirements only.

This platform contains the following hardware components:

- A crypto co-processor optimized for public key cryptographic calculations
- High speed Triple DES co-processor.
- High speed AES co-processor.
- PKI co-processor.
- Contactless Interface Unit fully compatible with ISO/IEC14443-4.
- USB 2.0 contact Interface according to ISO/IEC7816-12.[1]
- 160 Kbytes User ROM.
- 4608 bytes RAM.
- 72 Kbytes EEPROM for both data storage and program execution.

[1] USB interface is not supported in this module

4.2. Hardware Security Features

The following features are provided by the Philips P8WE5032 microcontroller.

- Low / high clock frequency sensors,
- Low / high temperature sensors,
- Low / high supply voltage sensors,
- Single Fault Injection (SFI) attack detection,
- Light sensors,
- Power-up / power-down reset,
- On-chip self test with signature technique,
- EEPROM erase / write timing independent from external clock,
- EEPROM erase /write operation controlled by hardware sequencer,
- Electronic fuses for safeguarded mode control,

These hardware security features were not tested as part of the FIPS 140-2 validation.

4.3. Physical structure

The die contains bonding pads for GND (ground), VCC (supply voltage), CLK (clock signal), RST (external reset signal), I/O 1 (used for half-duplex input / output communication), LA antenna coil connection, LB antenna coil connection, and I/O 2 (currently not used).

4.4. Fabrication Process

The die is attached with adhesive to the back of a film-based, ISO 7816-compliant contact substrate, such that the die bonding pads and the contact substrate bonding pads are accessible. Wire bonds are made between the die bonding pads and the contact substrate bonding pads for GND, VCC, CLK, RST, LA, LB and I/O. A measured amount of encapsulant is flowed over the contact substrate, such that the die and wire bonds are contained within the encapsulant.

The contactless antennas are then attached and the micro-module is glued into a matching milled cavity in a plastic card with the antennas embedded in it, which complies with the dimensional and other physical requirements of the ISO 7816-1 standard. Typical customization of the card for the application and / or user enterprise may be accomplished prior to and subsequent to micro-module insertion.

Silk screen (or other mass printing) and magnetic stripe application are performed prior to micro-module insertion. Personalization printing (such as names, account numbers, and photographs) and writing of the magnetic stripe data and personal data on the chip are performed after micro-module insertion.

5. Cryptographic Module Ports and Interfaces

5.1. Physical Interface

Five electrical connections are made between the die and the gold plated contact substrate of the smart card:

- **VSS**, Ground (reference voltage).
- **VDD**, Power supply input.
- **RST**, External reset signal from the interface device (card read / write device)
- **CLK**, External clock (3.517 MHz).
- **I/O**, Input or output for serial data to / from the processor.

The above five electronic signals are in full compliance with ISO 7816-3. The VPP (programming voltage) contact is not used because the EEPROM of the chip contains its own internal programming voltage generation. Also the I/O 2 and I/O 3 ports, which are present on the chip, are not connected to the contact substrate.

Antenna contacts are placed on the module backside antenna connections. In contactless mode Radio frequency power and signal interface are defined in ISO 14443 Parts 1 & 2, Type A communication.

Communication between the Proximity Coupling Device (PCD) and the cryptographic module is based on a standardized, half-duplex character transmission, ISO 14443 protocol, T= CL, defined in [ISO 14443-4]

Reference: ISO 7816 Part 3 Identification Cards – Integrated Circuit(s) Cards with Contacts – Electronic Signals and Transmission Protocols

5.2. Logical Interface

The SafeNet SCCOS is a file system based operating system that controls the logical interface thru a well-defined set of Application Protocol Data Unit (APDU) commands. It manages the secure object storage system, interface protocol and parameters, interprets and executes external commands.
The PIV application provides the functions required for FIPS 201 compliance and specified in special publication 800-73-1.
The cryptographic module submitted for this evaluation includes:

- SafeNet Cryptographic Card Operating System (SCCOS) Version 3.0.
- SafeNet PIV card application version 19.

5.2.1. SCCOS File System Architecture

SCCOS file system is hierarchical, with file types MF (master file), DF (directory file), and EF (data file). If the hierarchical file structure is represented as a tree, the MF is the root of

the file/directory tree. At manufacturing time, the card is initialized with the MF to contain two DFs and several vendor specific configuration EFs.
The first DF is the PIV application DF and the other is a general use application DF. The general use application DF contains SafeNet's PKI DF to be use with SafeNet's PKCS #11 middleware. The PIV application DF is the default selected card application.

5.2.2. SCCOS Security Architecture

The SafeNet Model 400 smart card implements an access control rule consist of an access type and a security condition called access right. The access type defines which operation can be performed on the data file such as Read, Write, Update and Delete and it is controlled by the permission bytes of each file.

Access Type	Definition
Read	File may be read by entity with ReadBinary command
Update	File data may be written by the entity up to the high water mark with WriteBinary or UpdateBinary commands
Write	File data may be written by the entity at the high water mark with WriteBinary command
Delete	File may be deleted by the entity with DeleteFile command
Execute	File may be used as a key file, a PIN file, or an ADF file
Resize	EF file size may be increased with the ResizeFile command
Recycle	DF may be recycled with the Recycle command

The lower nibble of each permission byte corresponds to an access right, which can take on the values 0 through 15. The 0 access right indicates that the permission is never granted. The 15 access right indicates that the permission is always granted. Access rights 1 through 14 can be defined in the Authentication Definition File (ADF) in the active DF.
In order for an ADF-defined access right to be activated, the authentication test in the ADF for that access right must be met successfully with the Verify command. The set of currently active access rights is available from the GetStatus command.

The security nibble definitions including the definitions of each type of access are described below:

Security Byte Array - EF					
Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**
Read	Update	Write	Delete	Execute	Resize

Security Byte Array - DF					
Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**
Read	Update	Write	Delete	Execute	Recycle

Security Byte Structure	
Bit 7	0 = Wireless mode allowed 1 = Contact mode required
Bit 6	0 = Secure messaging not required 1 = Ready State (secure messaging) required
Bit 5:4	PIN Expiration: 0 = No expiration/single-use 1 = Expires 2 APDUs after Verify, or when this access right used with this file. 2 = Expires 8 APDUs after Verify, or when this access right used with this file. 3 = No expiration – just single use for this access right with this file.
Bit 3:0	Access Right: *As defined in the Authentication section below.* 0 = Access never allowed 1-14 = Access allowed when this right is active. 15 = Access always allowed.

The contact-mode-required flag indicates that (for the associated file, access type, and access right) the access right is only good when communicating via a contact interface.

The ready-state-required flag indicates that (for the associated file, access type, and access right) the access right is only good when secure messaging is being used (Ready State).

The PIN Expiration flag indicates that (for the associated file, access type, and access right) the access right is only good for one usage after the matching Verify or ChangeReferenceData command, and then only within a limited number of APDU instructions after the Verify or ChangeReferenceData command.

The upper nibble of each permission byte in a file allows further control by defining whether the access right is valid during contactless communication, is valid without secure messaging (Ready State), and whether one-time authentication has expired.

The APDU communication protocol defines the following four logical interfaces as per the FIPS 140-2 standard as follows:

a) Data Input interface: The input data field of the command APDU comprises the data input interface of the module. All input parameters can only be passed through this interface.
b) Data Output interface: The output data field of the response APDU comprises the data output interface of the module. All output data can only be passed through this interface.
c) Control Input interface: The command APDU header consisting of the CLA, INS, P1, P2 and LC bytes comprises the control input interface. All control parameters for module execution can only be passed through this interface
d) Status Output interface: The status words SW1 and SW2 of the response APDU comprise the status output interface. All error codes and output indicators are output through this interface.

References: 1- SCCOS v3.0 Interface Control Document, March 23, 2006.

6. Roles, Services, and Authentication

6.1. Identity Based Authentication

The Model 400 smart card performs identity-based authentication using PIN, Biometrics and cryptographic keys. A specific access right is assigned to each PIN, cryptographic key and Biometrics to identify each role associated with the card. The following section describes roles and access rights assigned.

6.2. Roles

The SafeNet Model 400 smart card provides three independent roles, the Cryptographic Security Officer, Card Application Administrator and the User roles.

6.2.1. Cryptographic Security Officer Role

The Cryptographic Security Officer (SO) role is responsible for configuring the card by changing the configuration file settings (specifying which algorithms are allowed by the card, which keys may be generated, and who may generate keys), setting up the Card Application Administrator and User roles including defining the authentication data for these roles, setting the secure channel options. It is the SO's responsibility to ensure that the configuration file settings are set so that the module is in FIPS mode (see FIPS Mode section).
SCCOS initially authenticates the Cryptographic Security Officer role of the Root DF (Master File) by verifying a 20-byte long PIN that was set during chip fabrication. SCCOS automatically assigns access right 14 in the Master File directory to this SO role.

6.2.2. PIV Card Application Administrator Role

The Card Application Administrator Role (APP ADMIN) is responsible for creating and managing the PIV applications DF and set the security modes. This role is also responsible for generating key pairs and personalizing data objects as required by SP 800-73-1. SCCOS authenticates this PIV Card Application Administrator role by successfully executing External Authenticate challenge-response protocol using a 2-Key Triple DES key 9B.

6.2.3. User Roles

Card Holder User role

This is essentially the end user (USER) and thus has access to all of the cryptographic functions of the module, but does not have the access (that the Cryptographic Security Officer has or the Card Application Administrator) to the card configuration functions. SCCOS authenticates this User Role by verifying a PIN for the PIV application and verifying either a PIN , Biometrics, Challenge-response or a logical combination (AND, OR) of these methods for all other applications..

Card Holder unblocking role

This User role (USER UNBLOCK) is responsible for unblocking and managing the card holder PIN. SCCOS authenticates this User Role by verifying the PIN value.

Additionally, a few unauthenticated services are available. These services are listed in the table below. The services in the Idle (unauthenticated) state only provide general card status and do not provide access to cryptographic services or objects on the card.

State	Commands Available
Reset	None
Error	GetStatus
Unformatted	Format, GetStatus, TestRNG
Idle	EndSession, GenerateRandomNumber, GenerateSessionKey, GetData (object dependent), GetResponse, GetStatus, LoadEXF, SelectFile, SHA1, PackFileSpace
Ready	ChangeConfiguration, ChangeReferenceData, CreateFile, CreatePIVDF, Crypt, DeleteFile, DHKeyAgreement, DH/DSAGenerateKey, DSASign, DSAVerify, EnableDisableCIU, EndSession, FileInfo, FPEnrollTemplate, FPGetPublicTemplate, GeneralAuthenticate, GenerateAsymmetricKeyPair, GenerateDESAESKey, GenerateRandomNumber, GetData, GetResponse, GetStatus, LoadEXF, PackFileSpace, PutData, ReadBinary, Recycle, ResizeFile, RSADecrypt, RSAEncrypt, RSAGenerateKey, RSASign, RSAVerify, SelectFile, SHA1, UpdateBinary, Verify, WriteBinary

6.3. Services

6.3.1. SCCOS Services

Please refer to the ICD document for detailed information about each function including the required inputs and expected outputs

PIN, Keys and Fingerprint Management services

- **ChangeReferenceData**: Updates the PIN of the given type and other authentication data. After verifying that the given current authentication data is correct for the given access right, this command replaces the existing authentication data with the new authentication data supplied to this command.
  This command will fail within a PIV-II DF with "Instruction Not Supported" if using a contactless interface.
- **ResetRetryCounter:** This is basically an Unblock PIN command, but also works to unblock other authentication data. It works much like the ChangeReferenceData command, except that an unblocking PIN is used for initial verification instead of the old authentication data in ChangeReferenceData. After verifying that the given

unblocking PIN is correct for the given access right, this command replaces the existing authentication data with the new authentication data supplied to this command.
This command will fail within a PIV-II DF with "Instruction Not Supported" if using a contactless interface.

- **EndSession**: Ends the current authenticated access right session.
- **Verify**: Hashes the given data (operator pin) and compares the result with the value stored in the card's pin object container. If comparison is successful the module state is set to indicate successful authentication of the operator. When using this command for PIV-II card application PIN verification as described in NIST Special Publication 800-73-1.
  This command will fail within a PIV-II DF with "Instruction Not Supported" if using a contactless interface.
- **GeneralAuthenticate**: This implements the PIV-II GENERAL AUTHENTICATE command as described in NIST Special Publication 800-73-1. 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. The operation performed is context specific, and depends on the values passed in the data field. Following options are supported:
  Request Challenge: Include a zero-length challenge in the data field.
  Operation returns a non-zero length challenge in the response field.
  External Authenticate: Include a non-zero response to the previously obtained challenge data in the data field.
  No data is returned in the response. If successful, the associated access right is granted.
  Internal Authenticate: Include a non-zero-length challenge and a zero-length response in the data field. The challenge data will be encrypted (DES or AES) or signed (RSA), and the result returned in the response.
- **FPEnrollTemplate**: Creates the fingerprint template file on the token, and writes the initialized data to the file.
- **FPGetPublicTemplate**: Returns the public template data structure that is contained within the fingerprint template file on the token. The fingerprint template file must have been previously created with the FPEnrollTemplate command.

Cryptographic services

- **DH/DSAGenerateKey:** Generates a private exponent of the given length and computes the public/private key pairs for use by the Diffie-Hellman algorithm, the DSA algorithm, or both. DSA keys cannot be used in Approved mode.
- **DHKeyAgreement:** Completes a Diffie-Hellman key negotiation with the given DH public key y, the DH private key x in the given file, and the DH parameter p: negotiated_key = y^x mod p. The negotiated key is written to the command response field.

- **DSASign:** Performs a DSA signature on the given data with the private key and parameters in the given files. The returned signature consists of two 20-bytes values, r and s. DSA Sign cannot be used in Approved mode.
-
- **DSAVerify:** Performs a DSA signature verification on the given data with the public key and parameters in the given files. A return value of OK indicates the signature verified correctly, while a return value of Authentication Failed indicates the signature did not verify correctly. DSA Verify cannot be used in Approved mode.
- **Crypt**: Performs a DES/3DES/AES symmetric key encryption/decryption on the given data with the given direction, mode, and IV, using the algorithm and key found in the active symmetric key file.
- **GenerateAsymmetricKeyPair**: This implements the PIV-II GENERATE ASYMMETRIC KEY PAIR command as described in NIST Special Publication 800-73-1. Generates an RSA key pair into the files associated with the given key reference value. The private and public key files must already exist, either empty or containing existing keys. The key parameters supplied with this command must match the already existing files. If successful, the existing RSA key pair will be overwritten.
  This command fails with "Instruction Not Supported" if using a contactless interface.

- **GenerateDESAESKey**: Generates either a DES key (16 byte 2Key 3DES, or 24 byte 3Key 3DES), or an AES key (16 byte AES128, 24 byte AES192, or 32 byte AES256). It uses the on-card random number generator, storing the key in the active file.
- **GenerateRandomNumber**: Creates a random number of the given size, using the FIPS 186-2 (Appendix 3.1) compliant pseudo random number generator.
- **GenerateSessionKey:** Provides a means of negotiating a 2-key TRIPLE-DES or AES 128-bit session key with the host for secure messaging. Makes use of one-time Diffie-Hellman key agreement.
- **RSAEncrypt:** Encrypts the given plaintext with the public RSA key in the given file. If the Format Type is RSAPKCS1v1_5 the given plaintext is formatted according to the given PKCS #1 version type before being encrypted. Required to be issued in Secure Messaging mode.
- **RSADecrypt**: Decrypts the given ciphertext with the private RSA exchange key (or exchange/signature key) in the given file. Required to be issued in Secure Messaging mode.
- **RSAGenerateKey**: Generates an RSA key pair into the given private key and public key files.
- **RSASign**: Performs RSA PKCS #1 (version 1.5) signature on the given data with the private RSA signature key (or exchange/signature key) in the given file.

- **RSAVerify:** Performs an RSA verify operation on the given signature and hash with the public RSA key in the given file. If the Format Type is RSAPKCS1, the result is unformatted according to PKCS #1 before being compared to the hash.
- **SHA1**: Initiates, continues, or completes a SHA-1 hash of the given data.

Secure storage services

- **CreateFile**: Creates a file of the given type (EF or DF) in the active directory (DF).
- **CreatePIVDF**: Creates a PIV-II DF. This command is used only during the personalization process.
  The PIV DF is created in the current DF with a file ID of 0xFEDC, and a DF name of { 0xA0, 0x00, 0x00, 0x03, 0x08, 0x00, 0x00, 0x10, 0x00, 0x01, 0x00 }.
  The new PIV DF is set as the Default DF in the configuration file.
  The PIV Card Holder PIV Card Application PIN and its associated unblocking PINs are created and initialized.
  The PIV Card Application Administration Key is created and initialized as a DES or AES key.
  The PIV Authentication Key, PIV Card Application Digital Signature Key, and PIV Card Application Key Management Key are created as RSA keys, but are not initialized. These three keys may be initialized with the GenerateAsymmetricKeyPair command.
  All PIV data objects are created, but are left empty. The PIV data objects may be filled by using the PutData command.
  This command will fail with "Access Denied" if using a contactless interface.
- **DeleteFile**: Deletes the active file.
- **FileInfo**: Returns information about the active file.
- **GetData**: This implements the PIV-II GET DATA command as described in NIST Special Publication 800-73-1. Returns the data content from a single data object whose tag is given in the data field.
  If using a contactless interface, this command may be used only to get the PIV-II object "Card Holder Unique Identifier" (tag 5FC102).
- **PackFileSpace**: Packs the file system, restoring any file system resources deallocated with the DeleteFile command for use by the file system. The amount of file system space to be gained by packing can be determined from the GetStatus command.
- **PutData**: This implements the PIV-II PUT DATA command as described in NIST Special Publication 800-73-1. Completely replaces the data content of a single data object of a PIV-II card application with new content.
  command fails with "Instruction Not Supported" if using a contactless interface.

- **ReadBinary**: Returns the requested amount of data (at the given offset) from the active file.
- **Recycle**: Deletes all files and zeroes all allocated buffer space.
  This command will fail with "Access Denied" if using a contactless interface.
- **ResizeFile**: Resizes the active file to the given new file size.

- **SelectFile**: Makes the given file the active file, to be used by subsequent commands.
- **UpdateBinary**: Overwrites data in the active EF, at the given offset, with data given in the command.
- **WriteBinary**: Writes the given data (at the given offset) to the active file.

General Services

- **ChangeConfiguration**: Updates the current configuration data in the configuration file.
- **GetStatus**: Returns the current status of the card. The remaining file space gives the number of bytes available for creating new files (16-bit number, MSB first). The configuration data has the same format as the Configuration File.
- **Format**: During initialization of the card, this command is required to initialize containers storage system of a card and sets the initial SO PIN . This is a pre-issuance command and is not available once the card has been issued to the end user (Security Officer and User roles).
  This command will fail with "Access Denied" if using a contactless interface.
- **EnableDisableCIU**: Enables or disables wireless (contactless) communication with the card. This command shall not be used in FIPS mode to maintain PIV compliance. This command will fail with "Access Denied" if using a contactless interface.
- **GetResponse**: This function returns results from completion of previously executed GetData or FPGetPublicTemplate commands.
- **LoadEXF:** Attempts to validate the active file as an EXF. If the digital signature of the file data is valid, the EXF becomes an active EXF, enabling its contents for execution. Once an EXF file has been verified, it can be activated and deactivated in the future without performing the validation step. Once an EXF is activated, it remains active after a power/reset cycle.

The table below identifies which service available to each role.

Service	Role
ChangeConfiguration	SO
ChangeReferenceData	SO, USER
CreateFile	SO, USER
CreatePIVDF	APP ADMIN
Crypt	SO, USER
DeleteFile	SO, USER
DH/DSAGenerateKey	SO, USER
DHKeyAgreement	SO, USER
DSASign	SO, USER
DSAVerify	SO, USER
EnableDisableCIU	SO
EndSession	ALL
FileInfo	ALL
Format	SO
FPEnrollTemplate	SO, USER
FPGetPublicTemplate	SO, USER
GeneralAuthenticate	APP ADMIN, USER, SO
GenerateAsymmetricKeyPair	SO, USER
GenerateDESAESKey	SO, USER
GenerateRandomNumber	ALL
GenerateSessionKey	ALL
GetData	USER
GetResponse	USER
GetStatus	ALL
LoadEXF	SO
PackFileSpace	ALL
PutData	APP ADMIN
ReadBinary	SO, USER
Recycle	SO
ResetRetryCounter	USER UNBLOCK
ResizeFile	SO, USER
RSADecrypt	SO, USER
RSAEncrypt	SO, USER
RSAGenerateKey	SO, USER
RSASign	SO, USER
RSAVerify	SO, USER
SelectFile	ALL
SHA1	SO, USER

Service	Role
TestRNG	NA
UpdateBinary	SO, USER
Verify	SO, USER
WriteBinary	SO, USER

6.4. Authentication

As described in the SCCOS Security Architecture section 5.2.2, in order for an ADF-defined access right to be activated in the active sub-directory DF, the authentication test in the ADF for that access right must be met successfully with the Verify command.

Available methods include PIN, Challenge/Response, and Fingerprint verification and a logical combination (AND, OR) of these. The set of currently active access rights is available from the GetStatus command.

All access rights are application security based which means the status of the access right will be false when the currently selected application DF changes to another application DF.

6.4.2. PIV Card application Authentication

- The CreatePIVDF command creates a PIV-II DF. This command is used only during the personalization process.
- The PIV DF is created in the current DF with a file ID of 0xFEDC, and a DF name of { 0xA0, 0x00, 0x00, 0x03, 0x08, 0x00, 0x00, 0x10, 0x00, 0x01, 0x00 }. The new PIV DF is set as the Default DF in the configuration file.
- The PIV Card Holder PIV Card Application PIN and its associated unblocking PINs are created and initialized.
- The PIV Card Application Administration Key is created and initialized as a Triple DES key.

- The PIV Authentication Key, PIV Card Application Digital Signature Key, and PIV Card Application Key Management Key are created as RSA keys, but are not initialized. These three keys may be initialized with the GenerateAsymmetricKeyPair command.
- All PIV data objects are created, but are left empty. The PIV data objects may be filled by using the PutData command.
- The three sets of RSA keys, and any of the PIV data objects may be deleted when access right 0x03 is active. If any are not wanted, delete them as part of the personalization process.
- The PIN and keys are associated with File IDs and Access Rights as follows:

Key or PIN Name	PIN File ID	UB PIN File ID	DES/AES File ID	Private Key File ID	Public Key File ID	Access Right Granted
PIV Card Holder PIV Card Application PIN	FF00	FF01				1
PIV Authentication Key				A09A	A19A	2
PIV Card Application Administration Key			A09B			3
PIV Card Application Digital Signature Key				A09C	A19C	4
PIV Card Application Key Management Key				A09D	A19D	5

The PIV DF is created in the current DF as follows:

Name	File ID	File Type	Initialized	Security Permission Bytes					
				Read	Upd	Write	Del	Exec	Rec
PIV DF	FEDC	3800	Yes	0F	83	83	83	00	83

6.4.3. PIV Card application Authentication

Authentication Mechanism	Strength of Mechanism
PIN Verification	SCCOS allows digits, lowercase and uppercase letters and punctuation marks to be used as the 8-byte long PINs for PIV user authentication. Strength of this mechanism is therefore 1/80^8
General Authenticate	Using the external Authenticate option of the General Authenticate command, PIV Card Application Administrator authentication can be accomplished using the Triple-DES key 9B. Strength of this mechanism is 1/2^168.

6.4.4. General non-PIV Card Application Authentication

The cardholder must execute the Verify command with the correct PIN (an 8 byte secret) for PIV application or Verify with either the correct 8-20 byte secret, response (challenge encrypted using a TRIPLE-DES/AES key) or fingerprint verification data if a different application is selected other than the PIV application to transition the card state to User authenticated state. In this state, the User can access services provided by SCCOS that require User authentication.

To discourage an attacker from guessing the SO or User PIN or Challenge-response key and to discourage false acceptance from biometric verification, SCCOS maintains a count of the number of consecutive Verify (and ChangeReferenceData(User) attempts remaining (the limit is established by the ADF file) due to incorrect authentication data (PIN, encrypted challenge, or fingerprint). This count is maintained in nonvolatile memory. When this count reaches zero, the authentication method will be blocked, and an error response will be returned to the host. As defined by the ADF file, the authentication method can be unblocked to re-enable the method using a number of unblocking PINs stored in the unblocking PIN file. The Unblocking PIN also has its own count which is decremented on each incorrect PIN entry. When unblocking PIN count reaches 0, the unblocking PIN and authentication method are both blocked and can no longer be unblocked.

As for fingerprint, the number of allowed bad fingerprint authentication attempts is set when the fingerprint template is enrolled on the smart card. It can have one of the following values:

- 0: There is no limit to the number of bad attempts.
- 1: Only one bad fingerprint attempt is allowed. Two bad attempts will lock the fingerprint template.
- 2 through 63: 2 though 63 consecutive bad attempts are allowed.

The relatively high maximum limit (63) compared to the maximum smart card bad PIN limit (15) reflects the reality that a user is more likely to inadvertently mismatch on a fingerprint than a PIN.

The count of remaining bad fingerprint authentication attempts is kept internally on the smart card, and is independent of the internal bad PIN counter. It is decremented with every bad fingerprint logon attempt, regardless of which fingerprint is used. Switching fingers does not clear the count. The count of remaining bad fingerprint authentication attempts is set to maximum allowed by ADF with every successful fingerprint logon.

When the internal count of remaining bad fingerprint authentication attempts reaches 0, logon via the fingerprint template is locked. Once locked, no fingerprint can be used to log on until a new fingerprint template is enrolled onto the smart card.

A flag can be set during enrollment to lock this parameter. If locked, the maximum bad fingerprint limit is fixed and cannot be changed during future enrollments. Once the lock flag is set, it cannot be cleared during re-enrollment. It can only be cleared by deleting the entire fingerprint template via either the Recycle command or by deleting the cryptoki directory.

The following table summarizes the type of authentication and strength of mechanism for each role.

Role	Authentication	Strength of Mechanism
Card Holder & Card Holder Unblocking	PIN	8-20 characters 1/80^8 - 1/80^20
SO	PIN	8-20 characters 1/80^8 - 1/80^20
Card Holder	Fingerprint and PIN	At least 8-20 characters - 1/80^8 - 1/80^20 + biometric strength
Card Holder	Challenge-Response	Strength of this mechanism is equal to strength of key. For TRIPLE-DES 2-Key: 1/2^112. TRIPLE-DES 3-Key 1/2^168. AES 128-bit: 1/2^128 AES 192-bit: 1/2^192 AES 256-bit: 1/2^256

6.4.5. Security Relevant Data Item

The Security Relevant Data Items (SRDI) for the non-PIV DF of Model 400 smart card are:

1- Secure channel session key: TRIPLE-DES 2-Key or 128-bit AES key
2- Secure channel key exchange key
3- Card Holder PIN, 8-20 characters

4- Card Holder unblocking PIN, 8-20 characters
5- Card Holder Challenge-Response key: AES or TRIPLE-DES
6- Fingerprint template
7- Configuration file settings
8- Authentication Data Files (ADFs)

The SRDIs for the PIV DF are as follows:

1- PIV Card Application Administrator Key: 3-Key TRIPLE-DES.
2- PIV authentication RSA 1024-bit key
3- PIV card application RSA 1024-bit digital signature key
4- PIV card application RSA 1024-bit key management key
5- Card Holder PIN, 8-characters
6- Card Holder unblocking PIN, 8-characters
7- PIV authentication X.509 certificate
8- PIV card application digital signature X.509 certificate
9- PIV card application key management X.509 certificate

The SRDIs common to all applications are:-
1- SO PIN
2- PRNG 20-byte seed and seed key

The PINs can be considered Security Relevant Data Items (SRDI). However the module does not store any actual PINs in EEPROM. Only the SHA-1 hash of PIN value is stored in the User or SO PIN file, which cannot be read or written except by using the ChangeReferenceData command by an authenticated user that writes the hash of the new PIN to the PIN file. There is also a default SO PIN hash value stored in ROM, which is used to authenticate to the card for the first time.

The following table summarizes the SRDIs available to each role for non-PIV application:

Role	**SRDI**	**Type of access**
Idle	Configuration file settings	Read
	Authentication Data Files	Read
Card Holder	Internally generated secret and private keys	Use, Generate
	User loaded secret and private keys	Use/Write
	Internally generated public keys	Use/Read/Write
	User loaded public keys	Use/Read/Write
	Configuration file settings	Read
	Card Holder PIN	Use, Update
	Private Fingerprint Template	Use, Write
	Card Holder Challenge-Response key: AES or TRIPLE-DES	Use, Update
	Secure channel session key: TRIPLE-DES 2-	Use, Generate

	Key or 128-bit AES key Authentication Data Files PRNG 20-byte seed and seed key	 Read Use
SO	Configuration file settings Card Holder PIN Card Holder Unblocking PIN SO PIN Private Fingerprint Template Card Holder Challenge-Response key: AES or TRIPLE-DES Secure channel session key: TRIPLE-DES 2-Key or 128-bit AES key Authentication Data Files PRNG 20-byte seed and seed key	Read,Update Write Write Write, Update Write Write, Generate Use, Generate Write, Read PRNG 20-byte seed and seed key
Card Holder Unblocking	Card Holder unblocking PIN Secure channel session key: TRIPLE-DES 2-Key or 128-bit AES key Configuration file settings Authentication Data Files PRNG 20-byte seed and seed key	Use, Update Use, Generate Read Read PRNG 20-byte seed and seed key

The following table summarizes the SRDIs available to each role for PIV application:

Role	SRDI	Type of access
Idle	PIV authentication X.509 certificate PIV card application digital signature X.509 certificate PIV card application key management X.509 certificate	Read
SO	PIV Card Application Administrator Key Card Holder PIN Card Holder Unblocking PIN	Write Write Write
Card Holder	Card Holder PIN PIV card application RSA 1024-bit digital signature key PIV card application RSA 1024-bit key management key PIV authentication RSA 1024-bit key, PRNG 20-byte seed and seed key	Use, Update Use Use Use Use
Card Holder Unblocking	Card Holder unblocking PIN	Use
PIV Card Application Administrator	PIV Card Application Administrator Key: 3-Key TRIPLE-DES. PIV authentication key, PIV card application digital signature key, PIV card application bit key management key PIV authentication X.509 certificate PIV card application digital signature X.509 certificate PIV card application key management X.509 certificate PRNG 20-byte seed and seed key	Use Generate Write Write Write Use

6.5. Configuration File

The 400 has a configuration file (with file ID FF00). The configuration settings determine the overall security rules employed by the module. Only the SO is allowed to modify the configuration file settings by calling the ChangeConfiguration APDU. The first 40 bytes of data in this command contain the new configuration data, which is used to update the data in the configuration file. The second 40 bytes of data in this command contain the new configuration mask, which is used to update the mask in the configuration file:

The configuration file has two 40-byte sections - the first is the configuration data, and the second is a bit mask that specifies which bits of the first section may be changed by the SO. A '1' in a particular mask location indicates that the SO may change the corresponding bit in the configuration data, while a '0' indicates that the corresponding bit in the configuration data can not be changed.

The configuration data of the file has the following format

Data	Length in bytes
RSA Exchange enable/size	1
RSA Signature enable/size	1
DSA enable/size	1
DH enable/size	2
Public key formatting	1
Crypto command enable	2
Symmetric key enable	1
Secure Messaging/MF Recycle Control	1
EXF enable	2
PIV Control	1
Default DF Name Length	1
Default DF Name	16
RFU	7
UART-only feature disable bits	1
USB-only feature disable bits	1
CIU-only feature disable bits	1
Configuration Mask	40

The top nibbles of the RSA and DSA, and first DH enable/size bytes indicates the maximum allowable modulus size, and have the same definition as the public key length index of section 4.3. The bottom nibble of the RSA and DSA, and first DH enable/size bytes are defined below:

Bottom Nibble				
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Meaning**
0	x	x	x	No user readable private keys
1	x	x	x	User readable private keys allowed
x	0	x	x	User cannot generate keys
x	1	x	x	User may generate keys
x	x	x	0	User cannot load keys
x	x	x	1	User may load keys

The second byte of the DH enable/size field is the size of the maximum allowable private exponent, in bytes.

Public Key Formatting	
Bit 7	RFU
Bit6	0 - DH/DSA keys are not allowed 1 - DH/DSA keys are allowed
Bit 5	RFU
Bit 4	0 – DH Raw Format not allowed 1 - DH Raw Format allowed
Bit 3	RFU
Bit 2	RFU
Bit 1	0 - RSA Raw Disabled 1 - RSA Raw Enabled
Bit 0	0 - RSA PKCS1 Disabled 1 - RSA PKCS1 Enabled

The two Crypto Command Enable bytes control the availability of the crypto-related commands. A zero indicates the command is not available, while a one indicates that the command is available.

Crypto Command Enable - First Byte	
Bit 7	Crypt
Bit 6	DH/DSAGenerateKey
Bit 5	DHKeyAgreement
Bit 4	DSASign
Bit 3	DSAVerify
Bit 2	GenerateDESAESKey
Bit 1	RSADecrypt
Bit 0	RSAEncrypt

Crypto Command Enable - Second Byte

Bit 7	RFU
Bit 6	RSASign
Bit 5	RSAVerify
Bits 4:0	RFU

The upper nibble in the Symmetric Key Enable byte controls the use of Triple DES and AES (Crypt and GenerateDESAESKey commands) and the use of the ECB and CBC DES modes (Crypt command). The bottom nibble has the same definition as the bottom nibble of the public key enable/size bytes.

Symmetric Key Enable	
Bit 7	0 – Triple DES Disabled 1 – Triple DES Enabled
Bit 6	0 - AES Disabled 1 - AES Enabled
Bit 5	0 - ECB mode Disabled 1 - ECB mode Enabled
Bit 4	0 - CBC mode Disabled 1 - CBC mode Enabled
Bit 3	0 - No user readable symmetric keys 1 - User readable symmetric keys allowed
Bit 2	0 - User cannot generate symmetric keys 1 - User may generate symmetric keys
Bit 1	0 - AES Co-processor Disabled, use CPU for AES calculations. 1 – AES Co-processor Enabled
Bit 0	0 - User cannot load symmetric keys 1 - User may load symmetric keys

The Secure Messaging Control byte controls the enforcement of secure messaging.

Secure Messaging / MF Recycle Control	
Bit 7	0 – Command encryption is optional, and is based on the APDU class. 1 – Command encryption is required in Ready State. The APDU class must be set to 0x04 when in Ready State.
Bit 6	0 - Require Ready State for commands that specify it. 1 - Allow Ready State commands to execute in Idle State. (allows bypass of secure messaging)
Bits 5:4	RFU
Bit 4	0 – MF Recycle option #4 is not allowed. 1 – MF Recycle option #4 is allowed. (delete file IDs greater than 0x00FF, retain existing PIN).
Bit 3	0 – MF Recycle option #3 is not allowed. 1 – MF Recycle option #3 is allowed. (delete file IDs greater than 0x00FF, set fixed default PIN).
Bit 2	0 – MF Recycle option #2 is not allowed. 1 – MF Recycle option #2 is allowed. (reformat file system, retain existing PIN)
Bit 1	0 – MF Recycle option #1 is not allowed. 1 – MF Recycle option #1 is allowed. (reformat file system, set fixed default PIN)
Bit 0	0 – Access right for recycle permission must be active to recycle MF. 1 – MF may be recycled without access right for recycle permission.

The two-byte EXF field is interpreted as a 16-bit (MSB first) count that is one less than the minimum acceptable EXF (EXF's are identified with a 14 bit count). A value of zero indicates all EXF's will be accepted. A value of 0xFFFF indicates no EXF's will be accepted.

The PIV Control byte is defined as follows:

PIV Control	
Bit 7	0 – Enable PIV-II commands 1 – Disable PIV-II commands
Bit 6	0 – Allow PutData to directly write to data object file if temporary buffer space is not available. 1 – PutData returns error if insufficient temporary buffer is available.
Bits 5:0	RFU – set to zero

The 5 bytes of RFU space are reserved for use by EXF's. All bits and bytes labeled RFU are by default set to zero.

The Default DF Name Size and Default DF Name is the name of the DF which will be selected on power up.
If the Default DF Name Size is set to zero or the Default DF Name is set to any value which is not the name of an existing DF , the Master file is the DF which is selected on card power up.

The three bytes: *UART-only feature disable bits*, *USB-only feature disable bits*, and *CIU-only feature disable bits* all have the same bit definitions as follows:

Disable Bits Specific To Each Communication Method	
Bit 7	0 – Allow RSA functionality 1 – Disable RSA functionality
Bit 6	0 – Allow DSA functionality 1 – Disable DSA functionality
Bit 5	0 – Enable memory POST usage 1 – Disable memory POST usage
Bit 4	0 – Enable ROM POST usage 1 – Disable ROM POST usage
Bit 3	0 – Enable File System POST usage 1 – Disable File System POST usage
Bit 2	0 – Enable FameXE and DES POST usage 1 – Disable FameXE and DES POST usage
Bit 1	0 - Enable SHA-1 POST usage 1 - Disable SHA-1 POST usage
Bit 0	0 – Enable RSA and DSA POST usage 1 - Disable RSA and DSA POST usage

Note that USB interface is not supported in this module. The three bytes each come into play only when the appropriate communication method is being used. They are intended to allow the same card to be used as both a contact based high-security card (with it's expensive POST operation) and as a contactless card for building entry (without the expense of the mathematical POST operations).

6.6. FIPS mode:

For FIPS Mode, the configuration file should have the following values:

Field	Value	Meaning
RSA Exchange enable/size	0x77	RSA Exchange enabled; User readable private keys not allowed
RSA Signature enable/size	0x77	RSA Signature enabled; User readable private keys not allowed
DSA enable/size	0x00	DSA disabled
DH enable/size	0x77, 0xC0	DH enabled; User readable private keys not allowed; Maximum private exponent size is 192 bytes
Public key formatting	0x3F	All formatting enabled, DH-DSA keys not allowed
Crypto command enable	0xE7, 0xFF	DSA disabled; All other crypto commands enabled
Symmetric key enable	0xFF, 0xFF	Triple DES and AES enabled; AES co-processor enabled; ECB, CBC modes enabled
Secure Messaging / MF Recycle Control	0x45	Secure Messaging available, not required. Ready State Commands can run in Idle State. MF may be recycled without access right, Recycle option #2- reformat and keep existing PIN allowed.
EXF enable	0x00, 0x00	EXF's enabled
PIV Control	0x00	PIV-II Functions enabled; PutData can directly write to data object file
Default DF Name Length	0x00	The Master file is selected on power up
Default DF Name	All 0xFF	
RFU	All zero	RFU
UART-only disable bits	0x00	No features disabled specifically for UART mode
USB-only disable bits	0x00	No features disabled specifically for USB mode
CIU-only disable bits	0x00	No features disabled specifically for CIU mode

It is the SO responsibility to ensure the bit settings are set as per the table above when module is in FIPS Approved mode. Also the bit-mask in the configuration file must be set to all zeroes so that the configuration settings cannot be changed once set. An operator can use the GetStatus command to determine whether the module is in the Approved FIPS 140-

2 mode by matching the configuration file settings to the values mentioned in the table above.

For biometric verification, in order to be in a FIPS Approved mode, the SO or User enrolling the fingerprint template on the card (for authentication of Users) should ensure that False Acceptance Rate (FAR) value is set to at least 1 in 100,000 during enrollment using the SafeNet CIP utilities and that a PIN is also used to secure any file protected by biometric authentication.

7. Finite State Model

The state diagram for the 400 complies with, and in some instances is dictated by, the ISO 7816 and ISO 14443 standards for smart cards. The card's state diagram consists of several states as indicated in the SCCOS ICD document.
The loading of the PIV EXF and the settings of the default SO PIN takes place in the pre-issuance phase at the SafeNet facility using an HSM. The configuration file is then changed to disallow further loading of EXFs.

8. Physical Security

The SafeNet Model 400 Smart Card is a single-chip cryptographic module. It is designed to meet FIPS 140-2 level 3 requirements for physical security.

The 400 IC is a production quality IC. It meets commercial-grade specifications for power, temperature, reliability, and shock / vibration. It uses standard passivation techniques for the entire chip.

In addition to the passivation material, a coating covers the chip. The epoxy material fills a reservoir constructed around the die and wire bonds. The epoxy used on the back of the chip is resistant to solvents that are commonly available.

9. Cryptographic Key Management

SCCOS supports the use of cryptographic keys in four primary functions – authentication, digital signatures, key management and encryption.

9.1. Secure messaging Keys

GenerateSessionKey command triggers generating a session key with the host for secure messaging. It makes use of one-time Diffie-Hellman key agreement protocol to generate a session key. The Diffie-Hellman secret value used for this protocol is 160-bits.
If the algorithm is set to 3DES, a sixteen byte two-key Triple DES key is generated as the session key. If the algorithm is set to AES, a sixteen byte AES-128 key is generated as the session key. Secure messaging is activated after the command response is sent (since the host cannot compute the session key without the DH public key data in the response).

9.2. PIV application Keys

PIV application is created with the two types of keys 3DES and RSA keys. Only the 3DES Card application administration key contains an actual key value loaded during manufacturing of the card and to be used by the issuer entity. The RSA keys are created as empty buffers to be loaded using GenerateAsymmetricKey command.

Key Name	3DES	RSA Private andPublic Key	Access Right Granted
PIV Authentication Key		9A	2
PIV Card Application Administration Key	9B		3
PIV Card Application Digital Signature Key		9C	4
PIV Card Application Key Management Key		9D	5

9.3. Other non-PIV application Keys

User roles and Crypto-Officer roles in a non-PIV application can generate, load and use cryptographic symmetric and asymmetric keys with their respective NIST approved algorithms. Key sizes are listed in section 9.4.

9.4. PRNG seed and seed Key

These values are used in the FIPS 186-2 implementation of the PRNG. The PRNG state is stored in EEPROM and destroyed when card is recycled.

9.5. Key generation

The SCCOS generates the following types of keys:

- 128, 192- bit Triple DES
- 128- 256 bit AES
- 1024- 2048-bit RSA public and private keys
- 1024- 2048-bit Diffie-Hellman public and private keys

The module uses the FIPS 186-2 Appendix 3.1 and 3.3 compliant (SHA-1 based) PRNG to generate keys. A non-Approved, non-deterministic hardware RNG is used to provide additional entropy for the seed value used in the PRNG process. Additionally; 20-byte entropy is supplied by reading a value from the User Entropy File stored on the card.

Only an authenticated User can generate keys on the card. No internally generated secret or private keys can be read, written or updated in the FIPS mode. The SCCOS employs checks to make sure that the key generation functions GenerateDESAESKey, DH/DSAGenerateKey and RSAGenerateKey will not write to key files that are writeable/updateable/readable by anyone (including the User himself).

9.6. Key entry and output

The card allows only an authenticated User to perform cryptographic functions in FIPS mode. Thus only the operator owning the keys can use them for cryptographic purposes. . All secret and private key file security nibbles are checked during cryptographic operations to ensure that they are not updateable or readable by anyone. Secret and private keys may be loaded in key files and are protected by the security nibbles set by the User. Internally generated keys cannot be read or modified.

The user may load keys in key files in plaintext form. Keys can be protected against unauthorized modification, substitution and disclosure by setting the appropriate key file security nibbles.

9.7. Key Storage

The keys are stored in plaintext form in the card file system in a key file in EEPROM.
The User may also load keys in appropriate key files. These keys are also stored in plaintext.
As seen from Section 7 the User can set file permissions using three bytes of security nibbles during file object creation to allow only the authenticated Users to read, modify the public key value.

9.8. Key Zeroization

A key file can be deleted/zeroized by issuing the DeleteFile command. Additionally the Recycle command can be used to destroy all files including all module keys and PIN files.

10. Cryptographic algorithms:

The module only supports the following FIPS-approved algorithms:

- Triple DES (ECB, CBC modes-2key, 3key Triple DES) (Cert. #472)
- AES (ECB, CBC modes-128,192,256) (Cert. #455)
- SHA-1 (byte-oriented) (Cert. #519)
- RSA (1024-2048 bit modulus, PKCS#1 v1.5) (Cert. #174)
- RNG (FIPS 186-2, SHA-1 based) (Cert. #241)

The module also provides the following non-Approved algorithms:

- Diffie Hellman allowed in FIPS mode key agreement. This key establishment methodology provides between 80 and 112 bits of encryption strength
- RSA Encrypt/Decrypt can only be used within Secure Messaging for key transport purposes in FIPS mode. This key establishment methodology provides between 80 and 112 bits of encryption strength.
- DSA (non-compliant)

11. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The Model 400 module conforms to the EMI/EMC requirements specified in FCC Part 15, Subpart B, Class B.

The Model 400 cryptographic smartcard is a single chip and is a passive device. The reader/writer device shall supply operating power and oscillation frequency to the smartcard.

Refer to FCC certificate included in this documentation package labeled

EMC TEST REPORT
Declaration of Conformity for FCC
Part 15, Subpart B – Class B
(Personal Computer and/or Peripherals)
Report No: FR3181E

Lab Name: GARWOOD LABORATORIES, Inc.
Location: Pico Rivera, CA USA

12. Self Tests

The POST self test functions test the following critical systems: the RAM (internal and external), the FAME coprocessor, the file system integrity, and the checksum on the contents of the ROM. If any of these tests fail, the card will enter the Error state. The cause of the error can be determined with the GetStatus command.

Known answer tests are also performed for the following cryptographic algorithms: SHA-1, Triple DES, AES, PRNG, RSADecrypt, RSAEncrypt, RSASign, and RSAVerify. If any of these tests fail, the card will enter the Error state.

If a validated EXF has been loaded, its checksum is checked against a value stored in EEPROM. The EXF will also be examined to see if any of its functions are enabled as POST functions. .

The pairwise consistency test is performed each time a key pair is generated. If the key pair generated is used for Exchange only then only an RSADecrypt/RSAEncrypt is performed using the key pair. If the key pair generated is to be used only for digital signatures then simply an RSASign/RSAVerify is performed. If any of these tests fail, the card will transition to error state. No data is output from the module.

The **Continuous random number generator** test is performed each time a block of random data is requested from the FIPS 186-2 compliant PRNG.

If this test fails, the card will enter the Error state. The cause of the error can be determined with the GetStatus command. The operator must then issue the Recycle command to bring the module back to an operational state.

The module also performs a firmware load test by doing an RSA signature verification when an EXF is loaded on the card. In case this test fails the EXF is invalidated and removed from EEPROM

13. Security Guidance

13.1 Development Errors and Oversights

The primary object of concern relative to development errors and oversights is the SCCOS source code, and to a lesser extent the supporting software in the IT environment. The SCCOS source code is the primary concern because once the code has been embedded into the ROM of the smart card, it is not easy to change.

The steps taken during development of SCCOS to minimize errors and oversights were:

- use of cryptographic experts for both high-level and code design,
- frequent discussion among design group members of functional and performance objectives / specifications,
- frequent design reviews of design approaches,
- special code reviews with third party cryptographic experts,
- development of extensive test scripts,
- use of emulation equipment to evaluate OS code with the target IC,
- extensive testing of pilot production ICs with OS embedded in ROM, and
- extensive evaluation in customer trials.

There are currently no known errors or faults in the SCCOS Version 3 code.

13.2 User Guidance

When or before the card is issued, the end-user should be made aware that the card is an extension of the user's ID and is capable of generating a digital signature for the user, which is as valid and legal as a written signature on a paper document. For this reason the user should also understand that he /she should keep the card on their person or under lock and key when not in use, and to protect their secret pass phrase from observation when logging on.

At the time the card is issued, an initial user pass phrase is in the card. The issuer should be urged to immediately change the initial pass phrase to one which the user can easily remember but one which others cannot easily guess.

These things seem to be simple enough to remember, but in fact require some personal discipline on the part of the user. *Lapses in this discipline can lead to the use of an authorized user's card by an unauthorized user.*

1. Users must ensure that a new seed value is written to the User Entropy File before invoking key generation.
2. Users must ensure that secret and private key files created on the card have appropriate permission bytes set to prevent unauthorized modification or disclosure.
3. Users must ensure that Secure Messaging is not used with PIV application.
4. Users must regularly examine card for signs of physical tampering.
5. Users must ensure not to use RSA encryption and decryption for protecting data in order to be an Approved mode. RSA encrypt/decrypt should only be used for key exchange.
6. Users must ensure to use a minimum Diffie Hellman modulus size of 1024-bits in an Approved mode.

13.3 Protection against Unauthorized Users

If an unauthorized user can gain access to an authorized user's card **and** pass phrase, the imposter can act in every sense with all the capability as the true owner of the card. *This usurpation of the user's identity, at best would be very embarrassing, and at worst extremely costly, to the authorized user.*

Other than the above, there are no known vulnerabilities that can come from the end-user's lack of application knowledge or carelessness.

13.4 Security Officer Guidance

The issuing organization's senior security administrator may be responsible for ensuring that cards are issued with a card configuration file as well as subject / object / operation attributes in accordance with organization security policy, and the administrator is normally assumed to be trustworthy in fulfilling this responsibility.

If this assumption is valid and the administrator is well-trained and competent, there are no known vulnerabilities added as a result of the card issuing process.

However, if the security administrator is motivated maliciously, compromise of application security is possible.

There are a number of card configuration bits that are used to implement elements of the organization / application security policy. Some examples of malicious intent would be:

- The policy may state that the administrator cannot unlock 'locked user cards'. If the administrator sets this bit contrary to policy, he / she can collect the 'locked cards',

unlock and reissue them to unauthorized users, or use them personally for his own gain.

- The policy may state that cryptographic commands may not be processed from the Idle State. Setting this bit contrary to policy would allow unauthorized user with a card but no pass phrase to perform crypto operations.

To protect against such vulnerabilities and to ensure that the module is in an Approved mode of operation as defined in FIPS 140-2, the Security Officer must follow the following rules:

1) The Security Officer must set the configuration settings as per the table in Section 6.5. Also the bit-mask in the configuration file must be set so that the configuration settings cannot be changed once set
2) The Security Officer responsible of Creating and setting up the PIV application must ensure that the maximum number of PIN retries attempts meet FIPS 140-2 requirements.
3) The Security Officer responsible of Creating and setting up any application directory with an ADF file must ensure that the maximum number of PIN & Challenge-Response retries attempts meet FIPS 140-2 requirements. This value depends on PIN length and key size but a value of 10 will always meet the requirements.
4) The Security Officer or user enrolling the fingerprint template on the card for biometric verification should ensure that the FIPS 140-2 strength of authentication requirements is met by configuring the biometric enrollment settings correctly (See Section 6.6).
5) The Security Officer must create PIN files, key files and ADF files with appropriate permission bytes to prevent unauthorized disclosure or modification.
6) The Security Officer must change the default SO PIN as soon as the card is in possession of the SO and should not communicate his/her PIN to any entity.
7) When the Recycle command is issued the EXF file is deleted. This command should only be used when the card is destroyed.
8) The SO should unblock User PIN only for legitimate Users.

13.5 Potential Loss of Secure State

There are no known outsider scenarios or act-of-nature failures that leave the card in an insecure state, in which further attacks could more easily compromise the system security.